

SmartLicense: A Runtime Behavioral Authentication Approach for Software License Protection

Kuldeep Vayadande^{1,*}, Pallavi Mulmule², Jayamala Kumar Patil³, Dipti Yashodhan Sakhare⁴, Pramod Dhumale⁵, Anindita Khade⁶, Preeti Bailke¹ and Sangram Patil⁷

¹Vishwakarma Institute of Technology, Pune, India

²DES Pune University, Pune, India

³Bharati Vidyapeeth's College of Engineering, Kolhapur, India

⁴MIT Academy of Engineering, Alandi(D), Pune, India

⁵Bharati Vidyapeeth's College of Engineering, Navi Mumbai, India

⁶Bharati Vidyapeeth Deemed to be University, Nai Mumbai, India

⁷Bharati Vidyapeeth's College of Engineering, Pune, India

*E-mail: kuldeep.vayadande@gmail.com

Received for publication
April 21, 2026.

Abstract

Traditional software licensing methods rely on one-time activation using license keys, which are easily vulnerable to key sharing, unauthorized redistribution of software, and post-activation misuse. Once the software is activated, the system rarely verifies again for the authenticity of the license key holder. Some recent research in continuous authentication and behavioral biometrics focuses mainly on user identity verification and also on privacy preservation, but it does not extend to software license enforcement in a runtime environment. This paper proposes an approach for runtime behavioral authentication based software licensing, where the behavior of user is continuously monitored in the background during software execution. Instead of enforcing action solely on activation time, the proposed system evaluates behavioral patterns, such as frequency of usage, interaction characteristics, session consistency, and so forth are all used to generate a trust score. The license state is adaptively updated based on trust score, enabling detection of abnormal usage pattern or license misuse or sharing. The trust score is continuously updated in background to update the license state. The proposed approach emphasizes mainly on runtime monitoring, modular design and extensibility to avoid complex cryptographic overhead. By integrating continuous authentication into software licensing, the system will bridge a critical gap between authentication mechanisms and license enforcement. The system will enhance license security, will improve misuse detection, and support adaptive enforcement without degrading user experience. Further, no interruptions will be given in case where normal usage is detected. But, notifications will be given to user in case of detection of abnormal usage.

Keywords

software licensing, continuous authentication, runtime monitoring, software engineering

1. Introduction

Software licensing is a fundamental mechanism used in protecting intellectual property and regulating authorized use of software systems. Most of the commercial and academic software products available employ license keys or activation-based mechanisms for restricting unauthorized access. These mechanisms typically perform validation only once,

which is during installation or initial activation. Once the activation is successful, the software assumes continued legitimate usage and does not perform any further verification during execution. Several studies have found that such static licensing approaches are vulnerable to post-deployment misuse, which includes license key sharing, parallel execution across multiple devices, and unauthorized redistribution (Chaudhari et al., 2023). With the adoption

of distributed systems, cloud-based software, and remote working environments, the limitations of one-time license activation have become clearer. Software licenses can be reused across multiple devices it can be shared among users, or can be exploited through automated execution without violating the initial activation constraints. Studies on behavior anomaly detection in management and information systems have demonstrated that the normal usage patterns frequently occur after successful authentication, highlighting the insufficiency of static access control mechanisms (Sun et al., 2020). Existing license solution largely relies on server-side checks or static policy enforcement, which fails to capture fine-grained runtime behavior of software.

To overcome these drawbacks of static authentication, recent research has focused on continuous authentication, where system legitimacy is verified throughout runtime interaction rather than a single one-time activation point. Continuous authentication systems analyze behavioral characters, such as interaction patterns, session dynamics, and consistency of usage to enable runtime trust evaluation. Some approaches which go through behavioral biometric techniques have shown to improve security by detecting deviations from normal usage patterns after initial authentication (Baig et al., 2023). These approaches enable adaptive security decision based on observed runtime behavior rather than relying solely on static credentials.

Analyses of centralized access and authentication systems indicate that static trust models often fail to adapt to dynamic usage environments. Studies on single sign-on (SSO) and centralized authentication show that once the trust is established, the systems rarely verify again during execution, leaving them vulnerable to misuse scenarios that occur after login (Iskhakov et al., 2023). This limitation becomes particularly relevant for licensed software, where trust is granted at activation time but not reassessed later.

Despite significant advances in continuous authentication research, the application of software license enforcement remains limited. Past research mainly emphasizes the authentication of users, their privacy, and the accuracy of biometrics, using complex mathematical and cryptographic models. Thus, the software licensing process remains isolated from the process of authentication and monitoring.

The process of software licensing is becoming increasingly difficult in modern software development environments that are mostly distributed and

outsourced, in which centralization of licensing is difficult (Woo et al., 2023). Recent studies demonstrate that static authentication mechanisms can fail to detect abnormal user behavior occurring after a successful login, which leads to the growth of the idea of runtime behavioral analytics for continuous verification (Iskhakov et al., 2023).

Runtime monitoring is employed to monitor the behavior of systems in operation to ensure that they meet required operational criteria. This technique has found numerous applications, including anomaly detection, fault diagnosis, and security monitoring. Studies related to the detection of insider attacks and behavior analysis show that observing user behavior, system calls, and system utilization at runtime can help detect any anomalous or malicious activity that cannot be detected using static monitoring techniques (Sun et al., 2020). The use of runtime monitoring for continuous software license validation remains underexplored in the existing body of knowledge. Consistency in behavior has emerged as one of the key factors that indicate genuine usage of the system through continuous authentication. Various studies involving behavioral biometric analysis have revealed that although a user may exhibit variable behavior, the same often tends to stay within certain predefined limits (Baig et al., 2023). However, large-scale deviation from such behavior can be linked to activities, such as account sharing, automation, or even account compromise. The fact that behavioral consistency can be used to identify instances of misuse of a computer system without invalidating credentials opens up possibilities of applying this concept to software licensing.

This paper attempts to fill the research gap through a runtime behavioral authentication scheme in licensing software. The authentication scheme involves continuous monitoring of the behavior of the software while running the licensed software and determining its legitimacy through the trust metric approach. These include factors, such as concurrent sessions, consistent execution, and behavior patterns; these factors will be used to generate the trust score dynamically. In place of adopting a binary yes/no approach to determining the validity or otherwise of a license, the license can be enforced based on time trust evaluation.

The solution is based on a simple and rules-oriented approach that does not rely heavily on cryptography, making it easy to deploy in

practice as well as integrate into existing systems for authentication purposes. By focusing on runtime software behavior rather than explicit user identity, the approach minimizes privacy concerns while enabling effective detection of license misuse. The design therefore focuses on modularity, extensibility, and maintainability. Figure 1 shows organization of paper.

Also, the challenges were faced in complex supply chains in identifying and managing licenses. Studies show that accurate identification of license is critical to ensure legitimacy. Further, analytical approaches enable dynamic tracking of licenses which provides actionable insights for developers. These advancements show that there is a necessity for automation and a more intelligent license managing solutions (Jahanshahi et al., 2025).

With the increased cost of software licenses, many of the organizations are trying to adopt predictive models for optimization of license procurement and its utilization. Some techniques were forecasted such as ARIMA algorithm which have been successfully applied for analyzing historical license usage data upon which future demand can be predicted. So, by transforming stationary data into stable sequences and estimating model parameters, the organizations can generate early warnings if there is license shortages or underutilization. Such models can help in future planning of budget and help in reducing operational costs (Song et al., 2021).

II. Literature Review

SSO technology has become an important part of authentication mechanism. This mechanism allows users to access more than one system after a single login which leads to improvement in usability. This technology supports centralized identity management and enables organizations to gain control on role-based access. SSO implementations are classified commonly into enterprise and web-based models out of which web-based models utilizes protocols, such as SAML, OAuth, and OpenID for exchange of authentication data. The SSO technology lacks continuous verification during active sessions. Detection of anomalies in user behavior remains a critical challenge, particularly in systems having sensitive infrastructure. The authors proposed a machine learning approach which utilizes audit logs and browser fingerprints to identify their users and devices. The algorithm consists of three models namely—OneClassSVM, IsolationForest, and EllipticEnvelope to detect anomalies. For experimentation purposes, dataset of roughly 48,000 audit logs was used. The highest average accuracy achieved was approximately 78%, having EllipticEnvelope as 76% and OneClassSVM as 75% (Iskhakov et al., 2023).

One of the techniques used by the authors are by combining SSO with passkey-based passwordless authentication. To overcome challenges such as password fatigue, credential reuse, and phishing

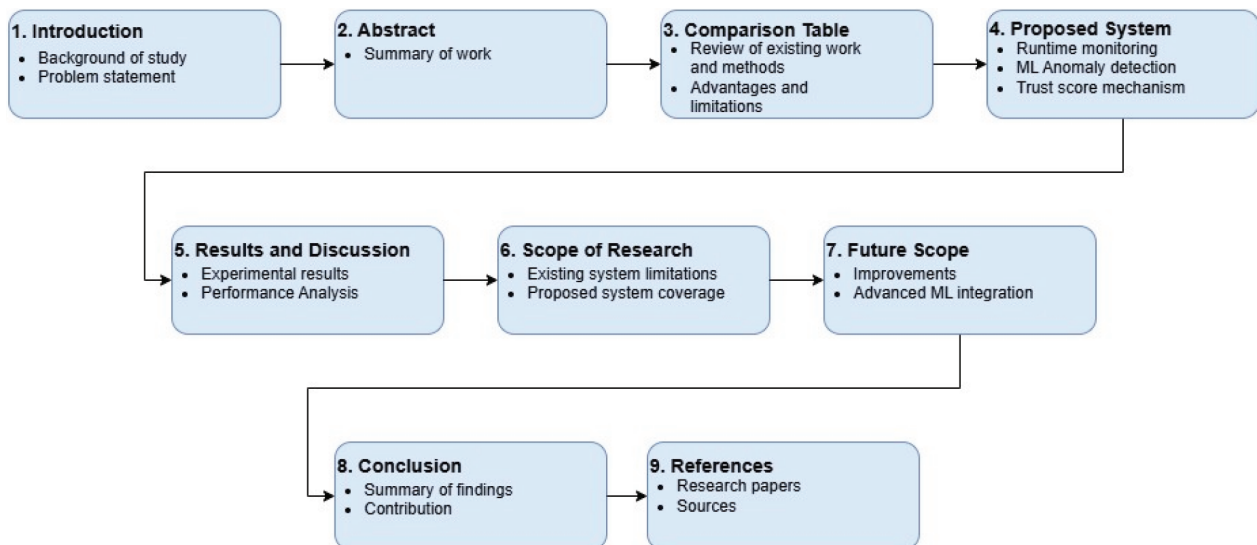


Figure 1: Organization of paper.

attacks, this study proposes a novel passwordless authentication framework that leverages passkeys, biometric verification, and cryptographic techniques. The study also suggests that while SSO improves usability and reduces repeated logins, there is a risk of session hijacking and insider threats. To overcome these issues, the paper also proposes an adaptive authentication model which aligns. The study concludes with combining SSO with passkeys and adaptive verification (Kamra and Shekhawat, 2025).

ElGamal encryption is a technique that is used to strengthen application security by detecting and preventing insider attacks. The authors highlight that insider attacks can pose significant risks because they can exploit legitimate credentials and sometimes network privileges. The proposed system incorporates an asymmetric key generation using ElGamal cryptography. The paper further introduces a password management server, service providers, and clients. A unique session key is generated in each transaction. At the end, the paper confirms that integrating ElGamal cryptography with session authentication can offer a reliable defense against insider attacks (Vemuri and Chaitanya, 2022).

Discussion of modern authentication systems, analyzing their advantages, vulnerabilities, and implementation, can provide a reliable base for authentication systems. The authors focus on traditional password systems, which introduce significant security risks, and the data may become vulnerable to hacking. The study reviews several mechanisms, such as single-factor, two-factor authentication (2FA), multi-factor authentication (MFA), and SSO. Further, the authors propose a basic passwordless authentication framework that utilizes biometric credentials delivered via mobile notifications after QR verification (Chaudhari et al., 2023).

There are growth challenges in software licensing in modern systems where open-source software (OSS) is commonly reused again and again. The study further investigates current practices in industry by analyzing both developers and legal practitioners. Further, automated tools such as Fossology and FOSSA assist in detection of licenses and their limitations. Overall, the work by the authors aims to streamline license workflows (Wintersgill, 2024).

Empirical investigation into license can conflict within free and open-source software (FOSS). The authors focus on widespread reuse of open-source components can increase license conflicts. The study further analyzes thousands of open-source projects to identify common patterns in conflicts. Overall, the work contributes to valuable empirical evidence

that supports the need for a more intelligent license mechanism (Cui et al., 2023).

OSS-LCAF is a framework which is designed to automatically identify and analyze license conflicts in open-source projects. The authors focus on development of modern software systems which highly depend on third-party packages. The proposed systems employ large language models (LLMs) to enhance license identification, which overcomes limitations of traditional rule-based approaches. Further, the framework proposed can distinguish between declared, inline, and referenced licenses, which enables more reliable detection of conflicts. Thus, the OSS-LCAF can be used for scalability and for an intelligent approach for automated license systems (Kahol et al., 2025).

Continuous authentication systems aim at improving the Android security on online examination platforms. The authors highlight that online exams are more susceptible to cheating when compared to traditional exams. This leads to the need for identity verification system. Further, the study focuses on multiple authentication approaches, which include knowledge-based, possession-based, and biometric methods. Out of these, biometric authentication provides stronger security. The system proposed captures images at random intervals using the normal device camera and stores the real-time data into a Firebase database. Overall, the research done contributes a practical approach for a secure online assessment environment that aligns with exam integrity (Aisyah and Subekti, 2018).

Continuous authentication approach for extended reality (XR) can be hybridized to integrate time-based one-time password, generally known as TOTP. Further, haptic feedback and kinetic activity monitoring can be used to enhance session security. The solution proposed operates in two phases: an initial login, which uses a haptic Morse code such as TOTP, followed by continuous verification based on rotational parity between device and head-mounted display (HMD). The system then monitors behavioral patterns and detect motion for detection of anomalies. The authors conclude that by combining behavioral patterns with tactile authentication can offer a strong defense (Grandi et al., 2023).

Large-scale datasets are designed to improve OSS license identification. The authors introduced a winnowing algorithm, which is a document fingerprinting technique. This technique is capable of detecting partial and embedded license texts even when formatting variations exist. Overall, the study provides a scalable approach (Jahanshahi et al., 2025).

A decentralized identifier (DID) system can be designed to enhance privacy, transparency, and traceability of software licenses. The authors say that outsourcing software development can introduce challenges. Therefore, a framework is proposed by integrating Restructured verifiable credentials (VCs) and software bill of materials (SBOM). By using this, one can manage licenses while protecting identity through zero-knowledge proof (ZKP). The authors conclude that DID technology can offer a reliable base for securing outsourced ecosystems (Woo et al., 2023).

LicenseRec is a knowledge-based system designed to assist developers in selecting appropriate open-source licenses. The authors highlight that choosing a suitable license is quite challenging due to the large number of OSI-approved licenses. Unlike the existing tools, LicenseRec considers real-world development and dependency constraints. A structured license is generated from 63 commonly used licenses, which encodes 19 key terms. These terms are related to copyright, patent grants, and redistribution constraints. Overall, the study suggests that knowledge-based systems can significantly reduce licensing errors (Xu et al., 2023).

A framework based on blockchain can be used to manage some of the history of copyright and license agreements in outsourced projects. The authors suggest that outsourcing a software project can be challenging due to transparency and copyright. To overcome this issue, a framework is used to integrate SBOM using the SPDX format with Hyperledger Fabric. This is used to record copyright and license agreements so that they become immutable. Overall, the study shows that a secure and scalable approach can be used for managing copyright and license agreements (Lee et al., 2023).

A comprehensive systematic literature review (SLR) that examines challenges, risks, and research trends. These are associated with OSS. The authors say that although there is OSS adoption, improper license handling can lead to exposure of organization. The authors propose future research directions that focus on standardized license frameworks (Li et al., 2025).

The authentication mechanisms used in microservices focus on the importance of secure communication in distributed systems. The authors highlight that microservices can improve scalability and flexibility, but can also introduce critical security challenges. The paper focuses on three mechanisms: API gateway authentication, then token-based authentication and mutual transport layer security (mTLS). The authors conclude that decentralized authentication

mechanisms are more suited for a microservices environment (Jack et al., 2023).

The ARIMA time series algorithm can be used to overcome the challenge of inefficient software license utilization in the petroleum industry. The authors highlight that inaccurate forecasting can lead to either license shortages or excessive procurement. The experimental study of ARIMA-based forecasting can achieve higher accuracy compared to traditional moving average methods. Overall, the study suggests that a data-driven forecasting approach can be used as a support for proactive license management (Song et al., 2021).

Intelligent performance evaluation has also seen some developments through deep learning-based sequence models. An example of such developments is presented by Wang, Nan, in their proposal of the PRO-BiGRU model, which combines poor-rich optimization and bidirectional gated recurrent unit to enhance the accuracy of performance evaluations in cloud computing platforms. The proposed model manages to capture complex sequences and attains an accuracy rate above 97% Wang (2025)

Latest studies have investigated intelligent and distributed anomaly detection methods in the context of network and IoT infrastructures. In their study, Villegas et al. (2025) presented an AI-based solution model incorporating deep learning algorithms with a distributed computing approach to detect and prevent spoofing attacks in IoT-based networks, showing that edge computing enhances the effectiveness and scalability of the anomaly detection process. Likewise, Arisandi et al. (2025) designed an "Invisible Scout" system that targets the detection of rogue access points in the Layer-2 environment, emphasizing real-time anomaly detection within wireless networks. The proposed method underscores the significance of analyzing network behavior at the lower layers and conducting continuous surveillance to detect any rogue access point successfully.

III. Comparison Table

IV. Proposed System

The proposed system introduces an adaptive runtime monitoring software license validation framework. Traditional license validation techniques primarily rely on a one-time validation method, such as entering the license key during initial login. However, these approaches are highly vulnerable to misuse, which includes sharing of license, unauthorized

duplication, concurrent execution across multiple devices, and some other forms of piracy. Since traditional authentication systems do not continuously monitor user behavior after initial validation, they fail to detect abnormal or suspicious usage patterns during runtime.

To address all these limitations, the proposed system adopts a dynamic and adaptive validation system. This system continuously monitors user activity and system behavior during execution of the software. Instead of solely relying on one-time or static validation, the system evaluates runtime parameters such as session frequency, execution duration, device identity, and some interaction patterns. These behavioral metrics are gathered in real-time and analyzed for any discrepancies from the usual patterns of usage. With the inclusion of runtime analysis, the process of authentication is ensured to be not just a one-off check but a continuous monitoring process. The temporal aspect is incorporated into the analysis through the grouping of run-time log data into fixed-length sliding windows, with each window denoting an activity sequence of recent user behavior. The use of sequences facilitates the detection of unusual behavioral patterns since several events are captured within a short period of time. This design involves an implementation of a hybrid approach for decision making using both machine learning methods to identify anomalies and the rules for validating results. The Isolation Forest method will be applied to find anomalies within the behavioral data. Then the result from the model will be combined with rule-based analysis, such as event rates, sessions, and location changes. The behavioral patterns that are found to be an anomaly could either result from the machine learning method or the rule-based approaches.

Moreover, this system incorporates a trust score algorithm that evaluates the users based on metrics during execution time. The user is judged based on several parameters, such as anomaly score, length of session, frequency of sessions, and consistency in the devices used. The parameters mentioned above are analyzed by a trust score engine to come up with the user's trust score. On the basis of this score, the policy engine decides on what course of action needs to be taken.

A key element of this system would be the binding and validation of licenses. The binding involves generating a unique device ID based on system information like MAC address and hostname, and then assigning it to each of the issued licenses. This way,

one cannot use the same license on more than one device. Centralized licensing validation helps scale the system and allows its usage in the real world. The system is based on metrics that have been extracted from the system at runtime, which include event occurrence, session occurrence, and device occurrence. These metrics eliminate the need for transforming textual content into numerical data using approaches such as TF-IDF.

For the effective implementation of the suggested system, the incremental development approach has been utilized. It allows for the creation of the system in several phases. In other words, it allows developers to test the runtime monitoring module, the anomaly detection module, the trust score evaluation module, and the policy enforcement module separately.

Though the system utilizes elements such as databases and visualization dashboards to implement the concept practically, the main focus of this project is on anomaly detection and adaptive trust-based validation. The configuration of the Isolation Forest algorithm took into consideration a contamination rate value of 0.1 to indicate the expected percentage of anomalies within the data. Other parameter values remained default, to preserve efficiency during computation. The set of features that served as the basis of training consisted of runtime metrics such as event rates and device usage.

In summary, the presented approach to adaptively license validate at runtime provides much more significant improvement in comparison to conventional static methods of authentication. As the adaptive licensing mechanism involves continuous monitoring and application of machine-learning anomaly detection and policy enforcement, it becomes capable of preventing software misuse and software piracy. Table 1 shows comparison table.

a. Dataset description

There is difficulty in detecting the anomalies within the dataset owing to the presence of very slight behavioral changes and a lower percentage of anomaly instances. It is very hard for the conventional models to differentiate between the anomalies and the normal ones. The dataset used in this work is composed of about 50,000 runtime logs obtained from simulations of software usage. These logs include various events including button press events, activities during the sessions, device use and geo-location details. Since there is no existing publicly accessible dataset on software license misuses, the creation of anomalies in this work was done through artificial means,

Table 1: Comparison table

Authors	Paper title	Technique used	Advantages	Disadvantages	Final remark
Iskhakov et al. (2023)	Enhanced user authentication algorithm based on behavioral analytics	Machine learning-based anomaly detection using audit logs and browser fingerprints	Improves authentication accuracy and device identification	Limited to the login phase and lacks runtime monitoring	Useful for anomaly detection but not suitable for software license enforcement
Kamra and Shekhawat (2025)	Enhancing user authentication with SSO and passkey integration	Passwordless authentication combining SSO and passkey-based verification	Reduces phishing risks and improves usability	Does not monitor runtime behavior	Suitable for secure login but not for continuous license validation
Vemuri and Chaitanya (2022)	Insider attack detection using ElGamal encryption	Cryptographic session authentication using ElGamal encryption	Strong protection against insider attacks	High computational complexity and no behavior monitoring	Focuses on encryption rather than license misuse detection
Chaudhari et al. (2023)	A comprehensive study on authentication systems	MFA using password, token and biometric verification	Improves authentication reliability	Does not support continuous monitoring	Useful for authentication but lacks runtime license enforcement
Wintersgill (2024)	Studying and improving software license compliance in practice	Automated tools for detecting open-source license compliance	Improves understanding of license compliance	No runtime validation mechanism	Focuses on license analysis rather than enforcement
Cui et al. (2023)	Empirical study of license conflict in FOSS	Empirical analysis of open-source license conflicts	Identifies patterns of license misuse	No automated prevention mechanism	Useful for research but not practical for runtime validation
Kahol et al. (2025)	OSS-LCAF license conflict framework	LLM-based automated license conflict detection	Intelligent detection of license inconsistencies	Does not monitor runtime software usage	Focuses on license identification rather than behavioral monitoring
Aisyah and Subekti (2018)	Continuous authentication for online exam systems	Image based biometric authentication during online exams	Provides continuous identity verification	Context specific to exam systems	Not applicable for software license validation
Grandi et al. (2023)	Continuous authentication for XR systems	Behavioral and haptic based authentication with TOTP	Enhances immersive system security	Limited to XR environment	Not designed for software licensing
Jahanshahi et al. (2025)	OSS license identification using winnowing algorithm	Document fingerprinting for license text detection	Detects embedded license texts effectively	Static analysis only	Does not validate runtime behavior

FOSS, free and open-source software; LLMs, Large Language Models; MFA, multi-factor authentication; OSS, open-source software; SSO, single sign-on; XR, extended reality.

using abnormal patterns that involve frequent events, using multiple devices, and making strange movements in terms of geo-location. The ratio of anomaly-to-normality was kept around 10%–15%. Figure 2 shows Architecture of the adaptive runtime software license validation system.

V. Results and Discussion

a. ML model metrics

Table 2 shows the comparison between previous studies and our proposed approach regarding the

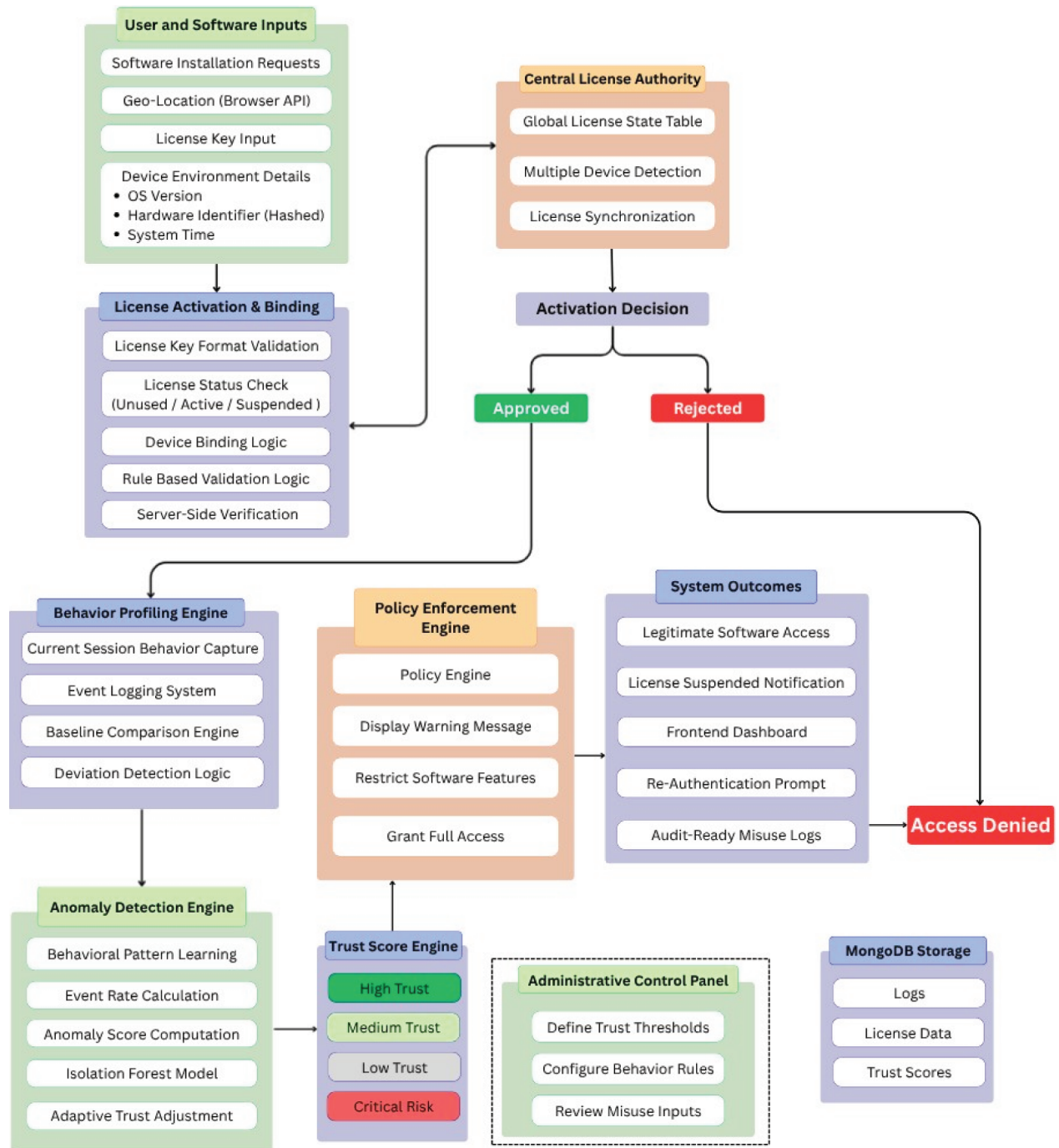


Figure 2. Architecture of the adaptive runtime software license validation system.

performance of the machine learning models used. As seen in previous research work by Iskhakov et al. (2023), Baig et al. (2023) and Aisyah and Subekti (2018), their models show remarkable accuracy when it comes to the topic of anomaly detection and authentication, with an accuracy between 88% and 94%. These methodologies have limitations concerning field-specific use or limited testing situations. However, the model that is suggested in our approach achieves an accuracy of 96.33%, along with high precision, recall, and F1-scores. It indicates the efficiency of the system in detecting the anomalies in runtimes. Higher recall values show better accuracy on the part of the system in detecting anomalies, while the F1 score indicates overall performance balance. As seen from the above F1 score, it seems that this score will be lower when compared to the score of the recall metric. The reason for this is the presence of imbalance in the nature of the data set. There were more normal logs in the dataset as compared to abnormal entries. In

such security-oriented systems, it is better to have higher recall scores than precision.

The baseline algorithms have been tested with respect to the same data set characteristics and also in their default settings. The relatively poor results obtained by them show how complex the runtime behavior data can be. Lower accuracy in terms of the behavior of the baseline models proves the complexity of behavioral data at runtime, as there are dynamic and non-linear elements involved. The drawback of using a single model alone proves the need to adopt a hybrid model approach. The experiment is performed by conducting an ablation study on each element separately to determine its impact on the effectiveness of the proposed model. The model performs moderately well without any additional elements, while incorporating rules increases sensitivity. The combination of both methods outperforms the others, proving the efficiency of the suggested structure. Figure 3 shows Machine Learning model Metrics.

Table 2: ML model evaluation metrics comparison

Model/study	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Iskhakov et al. (2023)	94	91	93	92
Baig et al. (2023)	92	89	90	89
Aisyah and Subekti (2018)	88	85	87	86
Proposed system	96.33	92.93	95.83	94.36

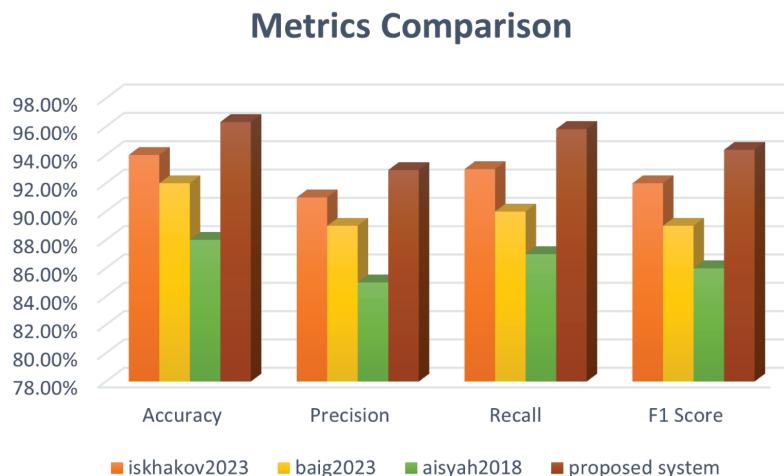


Figure 3. ML model metrics.

Table 3: Dataset size comparison (numerical)

Paper	Dataset size	Data type count	Sessions	Scale Index
Iskhakov et al. (2023)	1,600	2	1	5
Baig et al. (2023)	2	2	1	2
Aisyah and Subekti (2018)	500	1	1	3
Kamra and Shekhawat (2025)	300	1	1	2
Proposed system	1,000	2	1	4

b. Some other tables

Table 3 provides the comparison of sizes of datasets employed in different studies on authentication and behavioral analysis. The existing literature contains studies where the sizes of the datasets employed differed according to the study’s specific field. For example, Iskhakov et al. (2023) studied user activity log data containing 1,600 user records for detecting anomalies, hence improving the user behavioral model. Likewise, Baig et al. (2023) carried out continuous authentication by testing the effectiveness of two biometric datasets based on swipes and keystrokes. Additionally, Aisyah and Subekti (2018) conducted research about online exam systems and offered insights into behavioral modeling based on user activity data. The suggested system operates with an average-sized dataset containing around 1,000 session records. Figure 4 shows dataset size comparison.

Table 4 illustrates an analysis between various forms of authentication factors and techniques used in the selected research papers. Various researchers have attempted numerous ways in order to make authentication more reliable. To cite some examples, Iskhakov et al. (2023) employed three different machine learning classifiers called Isolation Forest, OneClassSVM, and EllipticEnvelope. In a similar vein, Kamra and Shekhawat (2025) designed an SSO system that is coupled with passkeys in order to use both password-based as well as passwordless authentication techniques. In addition to the above, Chaudhari et al. (2023) introduced MFA system employing passwords, tokens, and biometric identification.

Table 5 provides an overview of various security metrics detected in existing authentication techniques. Previous literature stresses the need to rectify any flaws associated with the authentication process. For example, according to Kamra and Shekhawat (2025), 80% of data breach cases occur due to credentials being stolen. In addition, Aisyah and Subekti (2018) found that about 73.6% of learners felt that

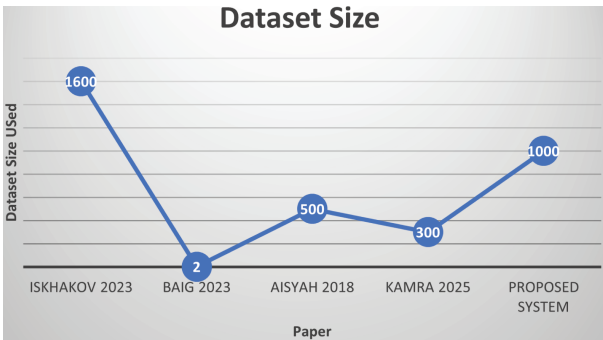


Figure 4. Dataset size comparison.

cheating was more manageable in online settings. Moreover, Iskhakov et al. (2023) illustrated how efficient anomaly detection was through extensive data sets. Therefore, the proposed system enhances these methodologies by adopting runtime detection and anomaly detection features, leading to better security results.

The Table 6 compares different authentication techniques used in existing studies. Behavioral anomaly detection approaches such as those proposed by Iskhakov et al. (2023) rely on machine learning models to analyze user activity patterns. Similarly, Baig et al. (2023) utilized continuous authentication based on behavioral biometrics such as swipe gestures and keystroke dynamics. Other approaches, such as Kamra and Shekhawat (2025), focused on passwordless authentication using passkeys integrated with SSO frameworks. In addition, Chaudhari et al. (2023) implemented MFA combining passwords, tokens and biometric verification. In contrast, the proposed system introduces an adaptive runtime validation mechanism that combines anomaly detection with trust score evaluation, resulting in improved efficiency and enhanced detection of abnormal software usage. Figure 5 shows authentication factor comparison.

Table 4: Authentication factors comparison (numerical)

Paper	Auth factors	ML Models	Security layers	Complexity Index
Iskhakov et al. (2023)	1	3	2	4
Kamra and Shekhawat (2025)	2	1	2	3
Chaudhari et al. (2023)	3	0	3	5
Baig et al. (2023)	1	1	2	3
Proposed system	2	1	3	4

Table 5: Security indicators comparison (numerical)

Paper	Breach risk (%)	Detection capability	Dataset size	Security score
Kamra and Shekhawat (2025)	80	2	300	4
Aisyah and Subekti (2018)	73.6	2	500	3
Iskhakov et al. (2023)	20	4	1,600	5
Baig et al. (2023)	25	3	2	3
Proposed system	10	5	1,000	5

Table 6: Authentication techniques comparison (numerical)

Paper	Technique count	ML usage	Runtime support	Efficiency score
Iskhakov et al. (2023)	1	1	0	3
Baig et al. (2023)	1	1	1	3
Kamra and Shekhawat (2025)	1	0	0	3
Chaudhari et al. (2023)	3	0	0	4
Proposed system	2	1	1	5

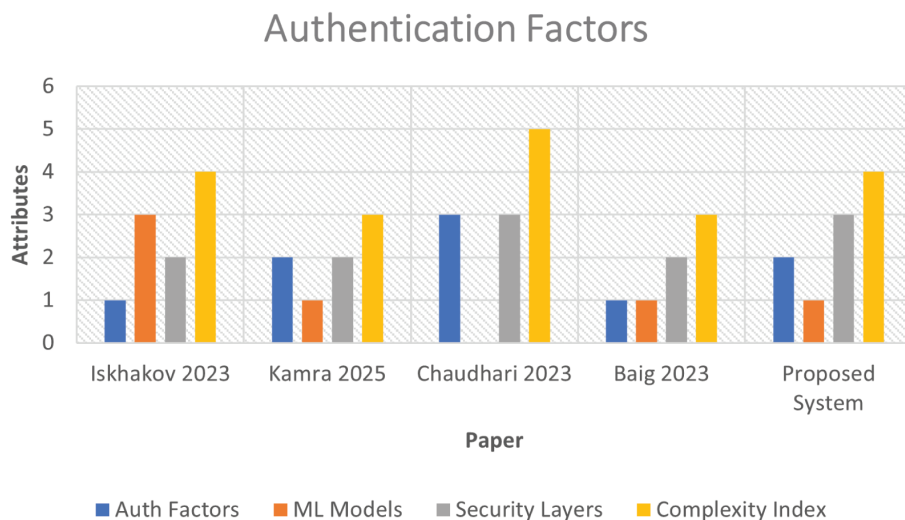


Figure 5. Authentication factors comparison.

Even though the number of logs analyzed in this evaluation is around 50,000, the dataset is adequate to verify the suggested approach. The system can easily scale up because of light-weight features and effective detection algorithms. It is due to the utilization of MongoDB that allows us to deal with larger databases. This system is built to execute in real-time, and uses fast feature extraction, alongside models that provide effective

anomaly detection in real time. The computation per request is extremely fast. MongoDB provides scalable storage and fast access to runtime logs. Figure 6 shows Security indicators comparison and Figure 8 shows License activation.

a. Implementation

a.i License activation

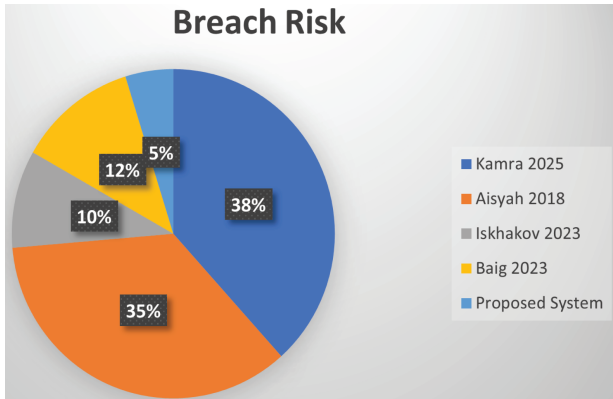


Figure 6. Security indicators comparison.

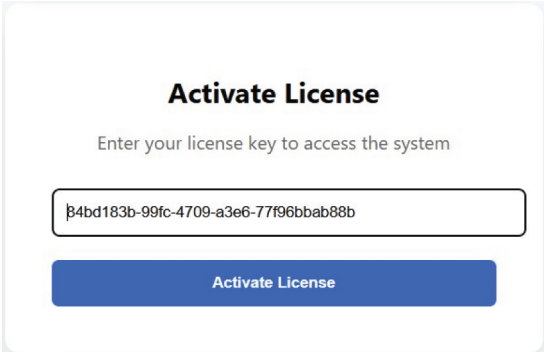


Figure 8. License activation.

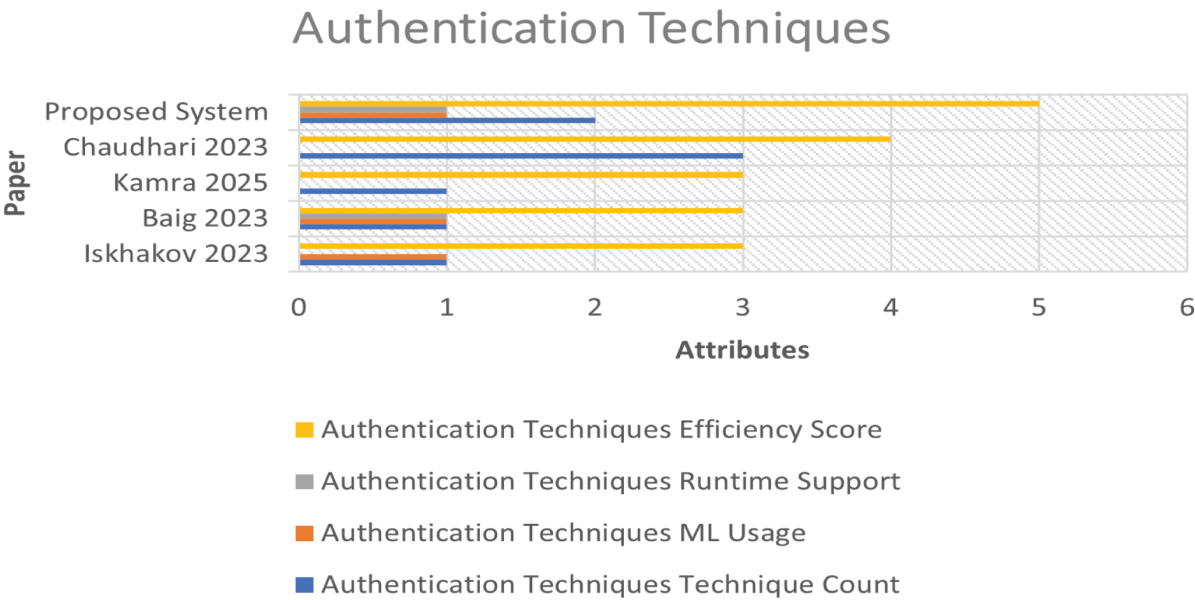


Figure 7. Authentication techniques comparison.

a.ii User panel

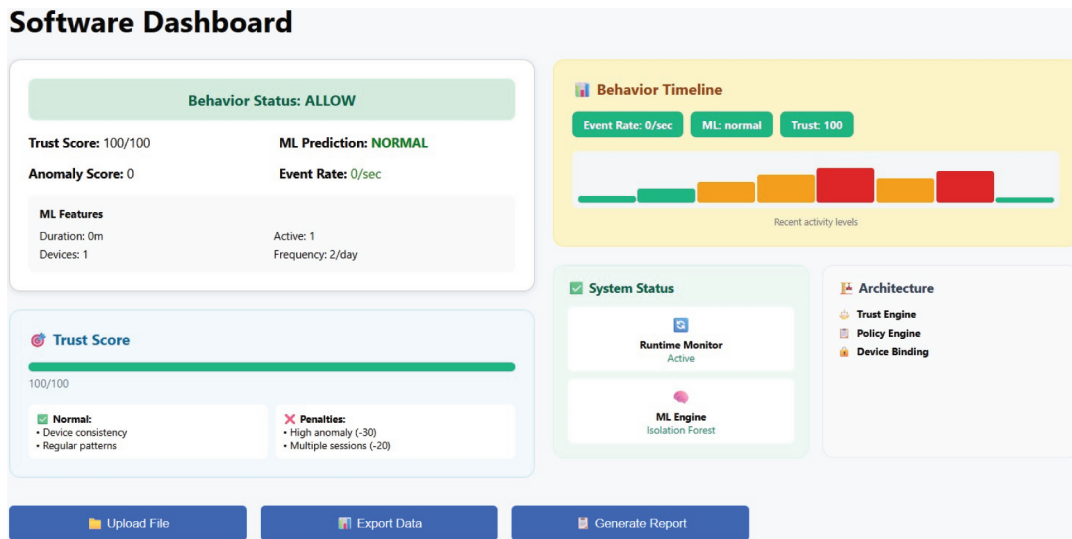


Figure 9. User panel.

a.iii Admin panel

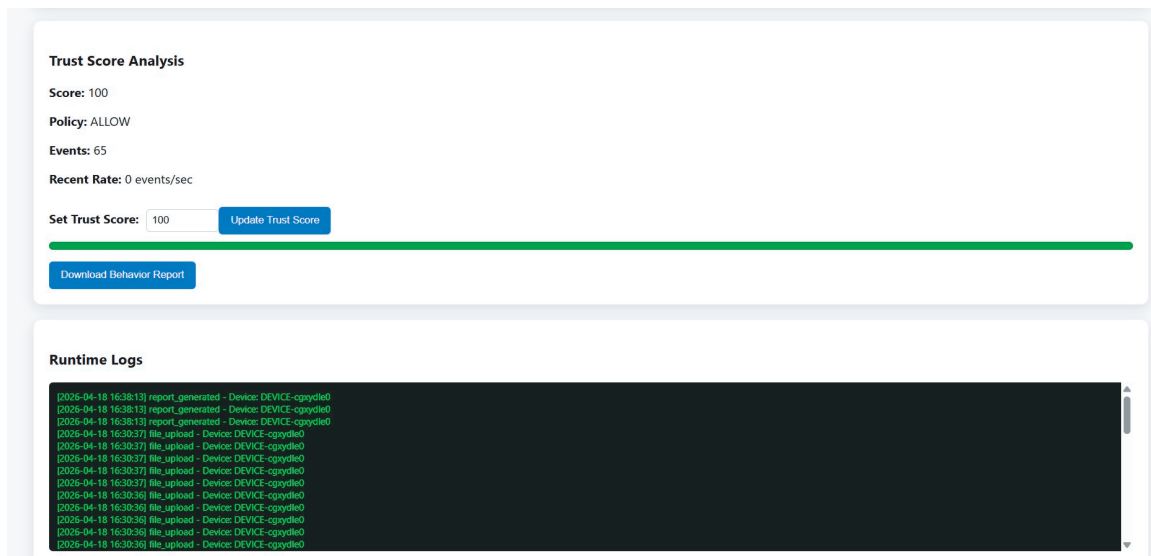


Figure 10. Admin panel.

Figure 9 shows User panel and Figure 10 shows Admin panel.

VI. Scope of Research

a. Existing systems

Current literature review in the area of authentication and software license validation mainly addresses

user identification during the first step of accessing the system. Some works like that of Iskhakov et al. (2023) use machine learning algorithms, such as Isolation Forest, One-Class SVM, and Elliptic Envelope for detecting abnormal behavior based on the user’s activities or browser fingerprinting. While the technique successfully identifies suspicious login attempts, its application ends when a user is granted

access; it does not monitor further runtime activities. Works by Kamra and Shekhawat (2025) and Chaudhari et al. (2023) concentrate on better authentication through SSO and MFA. Both methods involve a combination of different approaches to authenticate users such as usernames and passwords, and token/biometrics input. Though both contribute to enhancing the authentication process, they focus on securing access only and do not address issues regarding misuse after authentication.

Systems described in studies such as Baig et al. (2023) and Aisyah and Subekti (2018) implement continuous authentication techniques in specific domains such as online testing and biometric-based systems. This method employs behavioral biometrics such as keystrokes and swipes to authenticate users constantly. While extending conventional authentication schemes, these approaches have limited applicability and fail to incorporate the notion of software licensing.

Moreover, studies regarding software licensing like Wintersgill (2024), Cui et al. (2023), and Jahanshahi et al. (2025) emphasize license adherence, license conflict discovery, and license text analysis. While offering important perspectives in managing licenses, none suggest runtime protection mechanisms or abuse prevention techniques.

b. Proposed system

This system differs from the conventional software licensing system because it introduces an ongoing validation process beyond the first login process. The software license authentication systems available in the market only authenticate the software user at first and then assume that the user will use the software as intended in all future instances. However, the new system goes further and implements a continuous runtime check on the users' behavior when using the software.

Moreover, the system utilizes machine learning for detecting any anomalies in the user's behavior. In particular, it uses the Isolation Forest algorithm to train its anomaly detector. It can detect any anomalies in the user's behavior without having labeled data to train the algorithm. After detecting any anomaly, the system processes the information through a trust score engine to generate a dynamic trust score reflecting the user's trustworthiness. Depending on the trust score, the system decides whether to give full access, provide a warning, or restrict certain functionalities of the software.

Moreover, the system enhances the field of software security by adding multiple layers, such as device binding, behavior detection, and policy implementation within one architecture. Each software license will be uniquely bound with an exact device, thus, ruling out the possibility of its use on different machines. The use of run-time detection and decision-making processes make sure that not only is misuse detected, but actions are also taken based on the information. The result is that the system becomes highly appropriate for modern software usage models, such as SaaS models.

VII. Future Scope

There are numerous opportunities for further development and improvement in the proposed adaptive runtime license validation system. For instance, incorporating modern methods of deep learning into the algorithm will be beneficial for the detection of behavioral changes. RNNs and LSTM can track sequences and detect behavioral changes throughout time intervals much more accurately than the current model. Implementing the system on cloud-based architecture will allow centralized license management and scalable deployment. Furthermore, implementing geolocation analysis and network analysis can add another dimension of anomaly detection that will make the system capable of recognizing unusual license activity based on geolocations and network connections. Incorporating user profiling will increase the system's ability to identify unusual license activity. This procedure would include defining individual baseline behaviors for each user. Another potential application of the suggested adaptive license validation system at run time is using blockchain in the algorithm to store the license information in a decentralized manner. Extending the system to be compatible with other platforms, including IoT, mobile applications, and distributed systems, will enhance its adaptability in practice.

Possible future advancements will also include the creation of a self-learning or adaptive policy engine that continuously changes the threshold level and rules as per usage or emerging attacks. Besides, using a real-life large database to train the algorithm rather than using simulated training data will ensure that the machine learning algorithm is more robust and reliable. Improvements in the dashboard visualization and real-time analytics capability will allow administrators to have better insights into managing security. In summary, these improvements

will make the system smarter and more efficient for cybersecurity applications. The current system works on structured runtime logs. The future work will surely focus on handling variations in log format and template drift.

Although deep learning-based algorithms such as LSTM and RNNs have proven to be highly efficient in modeling sequences, this research concentrates on developing lightweight machine learning models to provide real-time processing capability. In future research, deep learning-based models will be studied for better pattern identification in time series analysis.

VIII. Conclusion

In this report, an approach to license validation is introduced that changes the paradigm from using static techniques to implementing an adaptive license validation technique based on runtime monitoring. Current approaches tend to consider only login-based user authentication without tackling potential license misuse during runtime. This paper introduces a system that uses runtime monitoring, anomaly detection based on machine learning techniques, trust score calculation, and adaptive policies to ensure effective and flexible license validation. The experiments carried out to assess the performance of the system show how accurate anomaly detection in the system is and the precision, recall, and F1-score values obtained are amazing.

In addition, a comparison with earlier methods shows that the system surpasses all other existing systems concerning issues of security, flexibility, and efficient monitoring. In conclusion, the current system can be considered a revolutionary breakthrough in the field of software security and licensing.

References

- Aisyah, S. and Subekti, L. B. (2018). Development of continuous authentication system on android-based online exam application. In *2018 international conference on information technology systems and innovation (ICITSI)*, pages 171–176. IEEE.
- Arisandi, D., Ahmad, N. M., and Kannan, S. (2025). Invisible scout: A layer 2 anomaly system for detecting rogue access point (rap). *Emerging Science Journal*, 9(1):284–310.
- Baig, A. F., Eskeland, S., and Yang, B. (2023). Privacy-preserving continuous authentication using behavioral biometrics. *International Journal of Information Security*, 22(6):1833–1847.
- Chaudhari, A., Pawar, A., Pawar, A., Pawar, A., and Pawar, G. (2023). A comprehensive study on authentication systems. In *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBE)*, pages 1–5. IEEE.
- Cui, X., Wu, J., Wu, Y., Wang, X., Luo, T., Qu, S., Ling, X., and Yang, M. (2023). An empirical study of license conflict in free and open source software. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 495–505. IEEE.
- Grandi, J. G., Terrell, J., Lofca, K., Ruizvalencia, C., and Kopper, R. (2023). A continuous authentication technique for xr utilizing time-based one time passwords, haptics, and kinetic activity. In *2023 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW)*, pages 959–960. IEEE.
- Iskhakov, A. Y., Mamchenko, M. V., and Khripunov, S. P. (2023). Enhanced user authentication algorithm based on behavioral analytics in web-based cyber-physical systems. In *2023 International Russian Smart Industry Conference (SmartIndustryCon)*, pages 253–258. IEEE.
- Jack, C. H., Teck, S. K., Ming, L. T., and Hong, D. Y. (2023). An overview analysis of authentication mechanism in microservices-based software architecture: A discussion paper. In *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, pages 1–6. IEEE.
- Jahanshahi, M., Reid, D., McDaniel, A., and Mockus, A. (2025). Oss license identification at scale: A comprehensive dataset using world of code. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 144–148. IEEE.
- Kahol, A., Tummepalli, A. C., and Anish, P. R. (2025). Oss-lcaf: Open-source software license conflict analysis framework. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 303–314. IEEE.
- Kamra, V. and Shekhawat, A. (2025). Enhancing user authentication with single sign-on and pass-key integration. In *2025 International Conference on Engineering Innovations and Technologies (ICoEIT)*, pages 495–499. IEEE.
- Lee, T., Lee, J., Woo, S., and Lee, J.-H. (2023). Copyright and license agreement history management framework for outsourced software. In *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, pages 690–691. IEEE.

Li, B., Liu, C., Fan, L., Chen, S., Zhang, Z., and Liu, Z. (2025). Open source, hidden costs: A systematic literature review on oss license management. *IEEE Transactions on Software Engineering*.

Song, M., Feng, M., and Miao, H. (2021). Petroleum software license usage forecast based on arima algorithm. In *2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC)*, pages 336–340. IEEE.

Sun, X., Yang, G., and Zhang, J. (2020). A real-time detection scheme of user behavior anomaly for management information system. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 1054–1058. IEEE.

Vemuri, S. A. and Chaitanya, G. K. (2022). Insider attack detection and prevention using server authentication using elgamal encryption. In *2022 International Conference on Inventive Computation Technologies (ICICT)*, pages 967–972. IEEE.

Villegas, W., Ortiz-Garces, I., and Govea, J. (2025). Ai-based architecture and distributed processing for

the detection and mitigation of spoofing attacks in iot networks. *Emerging Science Journal*, 9(3):1673–1693.

Wang, N. (2025). Pro-bigru: Performance evaluation index system for hardware and software resource sharing based on cloud computing. *HighTech and Innovation Journal*, 6(2):568–582.

Wintersgill, N. (2024). Studying and improving software license compliance in practice. In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, pages 225–227.

Woo, S., Lee, T., and Lee, J.-H. (2023). Decentralized identifier system for software copyright transfer and license management. In *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, pages 692–693. IEEE.

Xu, W., Wu, X., He, R., and Zhou, M. (2023). Licenserec: Knowledge based open source license recommendation for oss projects. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 180–183. IEEE.