e-ISSN: 2980-177X

JCEES

Volume: 3 Issue: 2 Year: 2025

Journal of Computer & Electrical and Electronics Engineering Sciences

MHA
MEDIHEALTH ACADEMY

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

CONTENTS

Volume: 3 Issue: 2 Year: 2025

ORIGINAL ARTICLES

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0026

An OWL-based ontology for modeling vocational school structures in educational systems

Selahattin Barış Çelebi[1], Bülent Gürsel Emiroğlu[2]
[1]Department of Computer Science, Faculty of Science, Engineering and Medicine, University of Warwick, Coventry, UK
[2]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kırıkkale University, Kırıkkale, Turkiye

Cite this article as: Çelebi, S. B., & Emiroğlu, B. G. (2025). An OWL-based ontology for modeling vocational school structures in educational systems. *J Comp Electr Electron Eng Sci, 3*(2), 35-40.

Received: 18.06.2025 ◆ **Accepted**: 30.07.2025 ◆ **Published**: 03.10.2025

ABSTRACT

Aims: The aim of this study was to develop an OWL-based ontology that accurately represents the structure of a university vocational school at the individual level, with semantic depth and logical coherence.

Methods: In this study, an OWL-based ontology representing the institutional structure of a vocational school affiliated with a university was developed. First, a domain analysis was conducted; core entities such as courses, programs, academic and administrative staff, and students, as well as their relationships, were identified. Then, this conceptual structure was formalized in OWL using the Protégé 5.6.6 editor. Object properties (such as teaches, learns) and data properties (such as student number, department) were defined to model inter-individual relationships and attributes. By adding individuals representing real-world scenarios, we ensured the instance-level applicability of the ontology. We validated logical consistency using the FaCT++ 1.6.5 reasoner. This approach supports both the conceptual integrity of the ontology and its potential integration into educational management systems.

Results: The developed ontology effectively modeled the components of the vocational school, such as courses, personnel, programs, and learning environments, at the class, property, and individual levels. Validation with the FaCT++ 1.6.5 reasoner confirmed logical consistency, and the inclusion of instance-level individuals demonstrated the ontology's potential for real-world application. The ontology provides a robust foundation for semantic querying and information extraction.

Conclusion: The developed ontology provides semantic consistency and real-world applicability in educational systems

Keywords: Ontology development, semantic web, web ontology language, educational information systems, vocational school modeling

INTRODUCTION

The rapid increase in web information volume has made traditional information access methods inadequate. This limitation reduces the effectiveness of data-driven decision support systems, especially in information-intensive environments such as educational institutions. In this context, semantic web technologies are emerging as a powerful paradigm that enables not only the presentation of data but also its interpretation (Patel & Jain, 2021). The semantic web aims to make existing web content semantically rich and machine-processable. Ontologies play a central role by formally and hierarchically defining relationships among concepts relevant to institutional structures (Türkyılmaz & Yaşar, 2008). Thus, content on the web becomes meaningful not only to humans but also to software agents.

In recent years, higher education institutions have increasingly needed ontology-based knowledge modeling due to the complexity of their internal structures, the large number of units, and the diversity of management processes (Emiroğlu, 2009). Multidimensional structures such as course programs, academic and administrative staff, and student profiles require a holistic representation of knowledge (Pavlidou et al., 2021). However, most studies in the existing literature only model the general structure of universities at an abstract level and do not give sufficient attention to concrete examples. As a result, ontologies lacking individual-level data often fall short in real-world educational applications.

This study aims to fill this gap in the literature by developing a detailed, individual-focused OWL-based ontology at the vocational school level of a university. It captures both conceptual and instance-level relationships across key academic units, including courses, programs, and personnel. Thus, it offers a structure that is applicable in real-world scenarios, supports inference, and is semantically rich. The ontology used in this study was modeled using Protégé and tested for logical consistency using the FaCT++ 1.6.5 reasoner. The ontology offers a semantic infrastructure suitable for integration into educational management systems.

Corresponding Author: Selahattin Barış Çelebi, sbariscelebi@gmail.com

RELATED WORK

Several studies have been conducted on ontology development for educational institutions, particularly universities. For example, one university ontology defined the overall conceptual structure but omitted specific instances such as students, courses, and faculty members (Ameen et al., 2012). Ontologies define the conceptual structure of universities, including classes, properties, and relationships, and can be adapted to represent institution-specific contexts. A similar effort was carried out using Protégé OWL, with a focus on the design rationale and modeling process (Malviya et al., 2011). In another study, a course ontology was constructed, emphasizing the development methodology and content structure (Zeng et al., 2009). While these studies make valuable contributions to the field, they generally focus on defining classes and properties and neglect detailed individual instances. For example, one ontology defined the structural framework of a university but did not include specific instances such as students, courses, or faculty members (Ameen et al., 2012). This limitation restricts the practical applicability of the ontology, as real-world applications require not only the schema but also populated data to perform meaningful queries and inferences.

In contrast, our work emphasizes the creation of a comprehensive ontology that includes detailed individual structures. By including specific examples such as student attributes and their relationships with courses and programs, our ontology supports realistic and operational use cases by incorporating detailed instance-level representations. This approach demonstrates how the ontology can be used in real-world scenarios such as student information systems or course registration platforms, thereby bridging the gap between theoretical ontology development and practical application. Our ontology was tested for logical consistency using FaCT++ 1.6.5, confirming its suitability for web system integration. By addressing the limitations of previous studies and developing an ontology enriched with instance-level data and logical consistency, this work advances the practical implementation of semantic web technologies in educational systems.

METHODS

In this study, an OWL -based ontology was developed to semantically represent the structure of a university's vocational school. The ontology was created using the Protégé 5.6.6 editor, and its logical consistency was verified using the FaCT++ 1.6.5 reasoner(*FaCT++ 1.6.5 Debian Package*, 2020; *Protégé 5.6.6 Ontology Editor*, 2025). The developed ontology consists of classes, object and data properties, and instance structures. In this section, the ontology development process, technologies, and methods used are presented in detail under subheadings.

Ontology Development Process

The development of the vocational school ontology was carried out using a systematic and phased approach. The process included the following steps:

Domain analysis: The organizational structure and operations of a vocational school were examined in detail, and the basic entities (e.g., courses, faculty members, students) and the relationships between them were identified.

Conceptualization: Based on the analysis results, a conceptual model containing classes and hierarchical relationships that form the basis of the ontology was designed.

Formalization: The conceptual model was translated into OWL language, and classes, object and data properties, and relationships were defined using the Protégé editor.

Sampling: Individuals representing real-world entities (e.g., a specific student or course) were added to the ontology to concretize the model.

Validation: The logical consistency of the ontology was tested using the FaCT++ 1.6.5 reasoner, inconsistencies were resolved, and implicit information was extracted.

During this process, challenges such as modeling the complexity of academic relationships and ensuring data consistency were encountered. These challenges were overcome through iterative improvements and discussions with domain experts. The developed ontology has enabled the vocational school structure to be formally represented in a machine-readable and semantically interpretable format.

Semantic Web

This is a technology that aims to make data on the internet understandable and processable not only by humans but also by computers (Berners-Lee et al., 2001). The evolution of the Web has progressed from Web 1.0 (static content) to Web 2.0 (user-generated content and interactivity), and toward the Semantic Web, which aims to embed meaning into web data for machine processing (Chada et al., 2025). While sometimes referred to as Web 3.0, the Semantic Web constitutes only one component of a broader Web 3.0 vision(Anwar, 2022). While content in Web 1.0 was provided solely by site owners, the rise of social networks and user contributions in Web 2.0 facilitated rapid information dissemination across the internet. However, the need for machines to process such large volumes of data laid the groundwork for the emergence of the Semantic Web (Can & Ünalir, 2010). The Semantic Web enables machines to interpret and process web data by embedding formal semantics-often through the use of ontologies and metadata-into the data itself (Ebietomere & Ekuobase, 2022). This allows search engines to provide more personalized and accurate results, marketing activities to be optimized according to user needs, and overall information access to be improved. The World Wide Web Consortium (W3C) has defined a layered architecture for the standardized development of the Semantic Web (**Figure 1**). This architecture encompasses data representation (RDF, OWL), querying (SPARQL), and trust mechanisms (Alesso & Smith, 2004). The trust layer, located at the top, emphasizes the critical role of information accuracy and reliability in the Semantic Web (Emiroğlu, 2009).

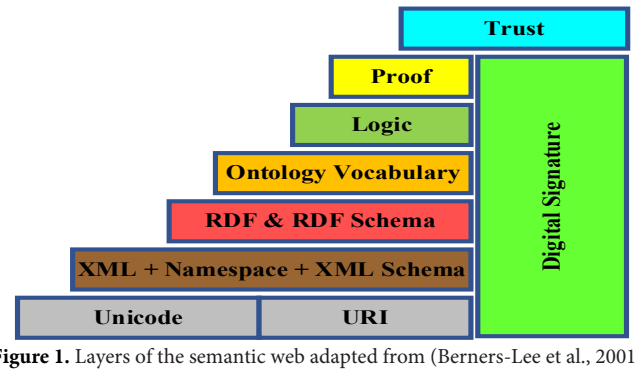

Figure 1. Layers of the semantic web adapted from (Berners-Lee et al., 2001)

Web Ontology Language

Ontologies are one of the fundamental building blocks of the Semantic Web and formally define the concepts, relationships, and constraints within a domain (Horrocks et al., 2004). OWL is an ontology language proposed by the W3C and extends the capabilities of the RDF (Resource Description Framework) and RDFS (RDF Schema) languages to enable the creation of more complex and meaningful structures (MICHAEL, 2004) . OWL supports advanced reasoning operations on ontological data by defining class hierarchies, property constraints, and logical axioms. OWL was developed as a successor to earlier ontology languages such as OIL and DAML+OIL, inheriting and formalizing their semantic expressiveness (Horridge et al., 2009).

In this study, OWL was chosen because of its widespread acceptance, strong tool support, and ability to model the complex relationships of an educational institution. Protégé 5.6.6 was used to develop the ontology, offering robust support for design, reasoning, and consistency validation through integration with FaCT++ 1.6.5. In the ontology, classes such as "Course," "Person," and "Program"; object properties such as "teaches" and 'learns'; and data properties such as "TC_ID_number" were defined. FaCT++ 1.6.5 reasoner was used to check the logical consistency of the ontology and extract implicit knowledge.

RESULTS

This section discusses the structure and components of the developed Vocational School ontology in detail. The ontology was created using OWL, designed with the Protégé 5.6.6 editor, and its logical consistency was verified with the FaCT++ 1.6.5 reasoner. The ontology consists of classes, object properties, data properties, and individuals. This section explains the ontology's class hierarchy, properties, and individuals, and discusses its potential contributions to semantic web applications for educational institutions.

Vocational School Ontology

The organizational structure of the vocational college has been modeled semantically in an open and formal manner. The ontology was developed to represent the fundamental elements of the institution and the relationships between these elements. Table 1 presents an overview of the ontology's hierarchical class structure, focusing on the primary components. It does not provide exhaustive details, but only summarizes the basic structure. The top-level class of the ontology is 'vocational school,' which is semantically decomposed into a hierarchy of subclasses representing institutional entities and roles.

The ontology models key vocational school components: courses, personnel, programs, and classrooms. The class hierarchy reflects the institutional structure, and subclasses inherit the properties of their parent classes. For example, the "person" class is divided into "academic," "administrative," and "student" subclasses, and these subclasses contain different properties and relationships. Each subclass is associated with distinct object and data properties that reflect role-specific attributes relationships.

Classes

Classes in the ontology represent groups of entities with similar characteristics (Noy et al., 2001). Organized in a hierarchical structure, classes express general concepts with superclasses and more specific concepts with subclasses (Taye, 2010). The ontology is structured around the top-level class 'VocationalSchool,' which semantically groups the primary subclasses: 'course,' 'person,' 'program,' and 'classroom.'

Course class: Covers courses offered in programs and is divided into the subclasses "required courses" and "elective courses." The 'course' class is associated with program-specific or cross-program offerings, modeled via object properties such as 'isOfferedIn'. For example, the course "asynchronous and synchronous machines" is only offered in the "electrical program," while "elective course-I outside the field" is offered in both the "electrical program" and the "gas and plumbing technology program."

Person class: Represents individuals within the institution and has the subclasses "academic," "administrative," and "student." Academic staff are responsible for teaching, while administrative staff handle managerial responsibilities. Individuals such as a director may belong to multiple subclasses, reflecting dual roles through multiple class memberships in the ontology.

Program class: The 'program' class includes educational offerings and is linked to departments through semantic relationships, distinguishing between institutional units and curricular components.

Classroom class: Represents the physical spaces where courses are conducted and is divided into subclasses such

Table 1. Vocational college ontology class structure

Top-level class	Sub-class	Instance/example
Vocational school	-	-
Course	Mandatory courses	Turkish language and literature I, power electronics I, asynchronous and synchronous machines, ...
	Elective courses	Information and communication technologies, non-departmental elective I, communication, ...
Person	Academic staff	Professor, associate professor, lecturer, ….
	Administrative staff	Director, deputy director, school secretary, department manager, clerk, …
	Student	First year, second year
Program	Department of electrical and energy	Electrical program, air conditioning and refrigeration technology program, …
	Department of electronics and automation	Electronic communication technology program, electronics technology program, …
Classroom	Laboratory	Computer laboratory I, chemistry laboratory, …
	Workshop	Electrical workshop, interior design workshop, …
	Lecture hall	Z01-Z06, auditorium I, auditorium II, …

as 'laboratory,' 'workshop,' and 'classroom.' This distinction ensures that courses are conducted in spaces appropriate to their content. The class structure is visualized in Figure 2.

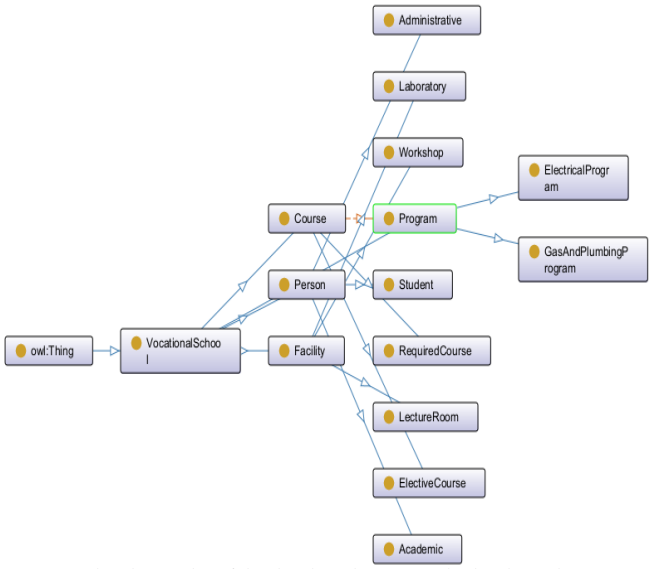

Figure 2. Class hierarchy of the developed vocational school ontology

Properties

OWL ontologies define two types of properties: object properties and data properties. Object properties are used to model relationships between classes, while data properties define attributes of individuals (McGuinness et al., 2004).

Object Properties

The object properties used in the study are listed below:

Teaches: Links an individual of the 'Academic' class to an individual of the 'Course' class.

Learns: Refers to the relationship between students and courses.

Cares: Represents a guidance relationship between staff members (academic or administrative) and students.

Registers: Defines the enrollment relationship from a student to a program. The involvement of staff in the registration process can be modeled via additional properties if necessary.

Learn lessons: Indicates the association between students and the classrooms where they attend lessons.

Teach lessons: It represents the relationship between academic staff and classrooms.

The 'teaches' object property links individuals from the 'Academic' class to those in the 'Course' class. In OWL, this is defined with the domain 'Academic' and range 'Course' as shown below:

- <owl:ObjectProperty

rdf:about="http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu# teaches">

- <rdfs:domain

rdf:resource="http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu# Akademik"/>

- <rdfs:range

rdf:resource="http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu#Ders"/>

- </owl:ObjectProperty>

Figure 3 shows the teaches object property, which links individuals from the academic Staff class to those in the course class, representing teaching responsibilities within the ontology.

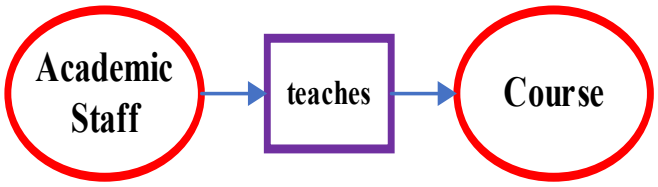

Figure 3. The Teaches relationship between academic and course objects

Figure 4 lists the core object properties defined in the ontology, modeling key relationships such as teaching, learning, registration, and guidance among academic entities in a formal semantic structure.

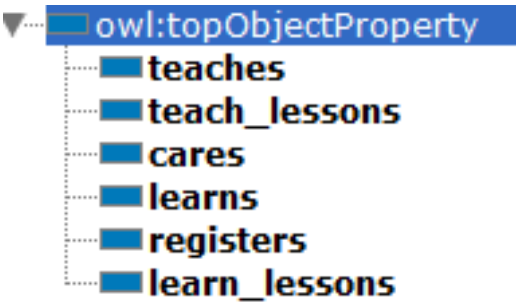

Figure 4. Object properties used in the ontology

Data Properties

This define attributes of individuals and are associated with specific data types. For example, the property "national ID" is a common attribute for the "academic", "administrative" and "student" classes and is of integer type, while "student number" is unique to the "student" class and can be defined as the following code block:

- <owl:DatatypeProperty

rdf:about="...#studentNumber">

- <rdfs:domain rdf:resource="...#Student"/>

- <rdfs:range

rdf:resource="http://www.w3.org/2001/XMLSchema# integer"/>

- </owl:DatatypeProperty>

These properties enable the representation of individual-level attribute data, supporting query and reasoning in semantic web environments. Table 2 presents the data properties defined in the ontology, specifying the attribute types associated with individuals from the academic staff, administrative staff, and student classes, thereby enabling instance-level semantic representation and reasoning.

Figure 5 shows the data properties defined in the ontology, created using the Protégé editor under the owl:topDataProperty hierarchy.

Table 2. Data properties and types used in the ontology

Academic staff	Data type	Administrative staff	Data type	Student	Data type
National ID	Integer	National ID	Integer	National ID	Integer
First name	String	First name	String	First name	String
Last name	String	Last name	String	Last name	String
Staff number	String	Staff number	String	Student number	Integer
Department	String	Department	string	Department	String
Academic title	String	Administrative title	string	Class level	String

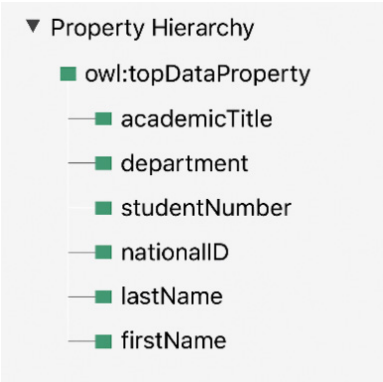

Figure 5. Data properties defined in the ontology

Individuals

Individuals represent concrete instances of classes and enable practical use of the ontology (Grimm, 2009). In the study, an instance of an individual named "Student_2" is presented:

- <!--

http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu#Ogrenci_2 -->

- <owl:NamedIndividual

rdf:about="http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu# Ogrenci_2">

- <rdf:type

rdf:resource="http://www.semanticweb.org/moem/ontologies/2018/0/meslek_yuksekokulu#1.Sınıf"/>

- <Ad

rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >Ahmet</Ad>

- <Soyad

rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >ÇELEBİ</Soyad>

- <Ögrenci_numarası

rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1702154532</Ogrenci_numarasi>

- <Bölümü

rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >Elektrik Programı</Bolumu>

- <Sınıfı

rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >1</Sinifi>

- </owl:NamedIndividual>

In OWL ontologies, individuals represent concrete instances of classes and enable the ontology to be used in real-world applications through instance-level data(Bouquet et al., 2003). The individual Student_2 is connected to eight Course instances via the 'learns' object property. In **Figure 6**, the 'learns' relationship between Student_2 and the course instance Direct_Current_Circuits is shown using an ontograph.

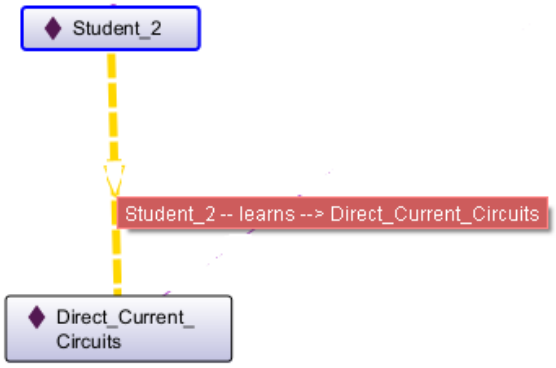

Figure 6. Object property assertion linking student_2 to direct current circuits via 'learns

Figure 7 illustrates the 'learns_lessons' object property between Student_2 and the course Direct_Current_Circuits, indicating which course the student attends.

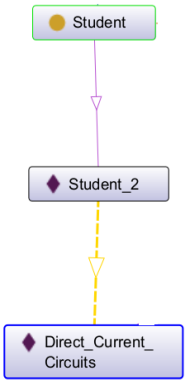

Figure 7. Object property assertion linking student_2 to direct_current_circuits via 'learns_lessons

In this OWL-based ontology, the relationships between classes and properties are modeled hierarchically from general to specific. The top-level class is vocational school, which includes key subclasses such as person, program, course, and classroom. The person class is further divided into academic, administrative, and student subclasses. Within the student class, year levels are represented as subclasses, including first year, to which the individual Student 2 belongs. Student 2 is enriched with data properties such as name, ID number, department, and class level. It is connected to eight course individuals via the learns object property, and to three classroom instances via learns lessons. In addition, the figure illustrates its inheritance from higher-level classes and associations with program and department entities. The semantic structure and hierarchical positioning of Student_2 are shown in **Figure 8**.

DISCUSSION

The developed ontology formally represents the organizational and functional structure of a vocational college in a machine-

readable and semantically interpretable format. Through a clearly defined class hierarchy, object and data properties, and richly populated individuals, the ontology captures complex institutional relationships in a formal manner. This enables enhanced semantic interoperability, supporting applications such as student information systems and academic resource management platforms. Unlike studies by (Ameen et al., 2012) and (Malviya et al., 2011), which focus on class and property definitions, our ontology emphasizes individuals, enhancing its practical applicability in educational contexts. Future work may include integrating the ontology into broader educational ecosystems and establishing data sharing frameworks based on Semantic Web standards such as RDF and Linked Data principles.

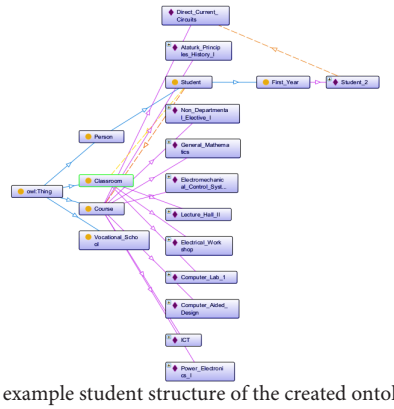

Figure 8. An example student structure of the created ontology

CONCLUSION

In this study, an OWL-based ontology was developed to formally represent the structure and semantics of a university vocational school using the Protégé 5.6.6 ontology editor. The domain of the vocational school was analyzed, and key concepts were identified and modeled as OWL classes. A class hierarchy was constructed, relevant object and data properties were defined, and individual instances were created to reflect real-world entities within the institution. Unlike many previous studies, this work placed emphasis on detailed instance-level modeling, enhancing the ontology's practical applicability. The ontology was validated for logical consistency using the FaCT++ 1.6.5 reasoner, and no logical inconsistencies were detected. The resulting structure is suitable for integration into web-based educational systems, supporting semantic representation and querying. The Semantic Web, as an evolving technological paradigm, provides a foundation for machine-interpretable content through ontologies. By embedding meaning into data and establishing formal relationships among concepts, ontologies enable intelligent applications to perform reasoning and enhance interoperability across systems.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Alesso, H. P., & Smith, C. F. (2004). Developing Semantic Web Services (0 ed.). A K Peters/CRC Press.

Ameen, A., Khan, K. R., & Rani, B. P. (2012). Construction of university ontology. *2012 World Congr Inform Commun Technol*, 39-44. doi:10.1109/WICT.2012.6409047

Anwar, A. A. (2022). A survey of semantic web (Web 3.0), its applications, challenges, future and its relation with Internet of things (IoT). *Web Intell*, *20*(3), 173-202. doi:10.3233/WEB-210491

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). Web semantic. *Scient Am*, *284*(5), 34-43.

Bouquet, P., Giunchiglia, F., Van Harmelen, F., Serafini, L., & Stuckenschmidt, H. (2003, October). C-owl: Contextualizing ontologies. In International Semantic Web Conference (pp. 164-179). Berlin, Heidelberg: Springer Berlin Heidelberg.

Can, Ö., & Ünalir, M. O. (2010). Ontoloji tabanlı erişim denetimi. *Pamukkale Univ J Eng Sci*, *16*, 2.

Chada, L. M., Parashar, H. V., Singh, I., Rastogi, M., Singh, S., & Raja, S. P. (2025). A comparative analysis of Web 3.0, Web 2.0, and Web 1.0: evolution and implications. *Int J Learn Change*, *17*(1), 1-55. doi:10.1504/IJLC.2025.143530

Ebietomere, E. P., & Ekuobase, G. O. (2022). Semantic web technologies. In Semantic web technologies (pp. 1-24). CRC Press.

Emiroğlu, B. G. (2009). Revolution on the web based educational technologies: Semantic web web tabanli eğitim teknolojilerinde devrim: anlamsal ağ. Proceedings of 9 Th International Educational Technology Conference.

Emiroğlu, B. G. (2009). Semantic web (anlamsal ağ) yapıları ve yansımaları. *Akademik Bilişim*, *9*, 151-155.

FaCT++ 1.6.5 Debian package. (2020, October). https://tracker.debian.org/pkg/fact++

Grimm, S. (2009). Knowledge Representation and Ontologies. In M. M. Gaber (Ed.), Scientific Data Mining and Knowledge Discovery (pp. 111–137). Springer Berlin Heidelberg.

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2009). A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2. *Univ Manchest*, *107*.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: a semantic web rule language combining OWL and RuleML. *W3C Member Submiss*, *21*(79), 1-31.

Malviya, N., Mishra, N., & Sahu, S. (2011). Developing university ontology using protégé owl tool: process and reasoning. *Int J Scient Eng Res*, *2*(9), 1-8.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C Recommendation*, *10*(10), 2004.

Michael, C. (2004). Owl web ontology language guide. http://www. w3. org/TR/owl-guide/.

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: a guide to creating your first ontology.

Patel, A., & Jain, S. (2021). Present and future of semantic web technologies: a research statement. *Int J Comput Appl*, *43*(5), 413-422. doi:10.1080/1206212X.2019.1570666

Pavlidou, I., Dragicevic, N., & Tsui, E. (2021). A multi-dimensional hybrid learning environment for business education: a knowledge dynamics perspective. *Sustainability*, *13*(7), 3889. doi:10.3390/su13073889

Protégé 5.6.6 ontology editor. (2025, May). https://github.com/protegeproject/protege/releases/tag/5.6.6

Taye, M. M. (2010). Understanding semantic web and ontologies: theory and applications. *arXiv Preprint arXiv:1006.4567*. doi:10.48550/ARXIV.1006.4567

Türkyılmaz, İ., & Yaşar, C. (2008). Semantik web teknolojileri. *Akademik Bilişim 2008*. 325-331.

Zeng, L., Zhu, T., & Ding, X. (2009). Study on Construction of University Course Ontology: Content, Method and Process. 2009 International Conference on Computational Intelligence and Software Engineering, 1-4.

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0027

Comparison of machine learning classification models for predicting student academic performance

Mahmut Ünver
Department of Computer Technologies, Kırıkkale Vocational School, Kırıkkale University, Kırıkkale, Turkiye

Cite this article as: Ünver, M. (2025). Comparison of machine learning classification models for predicting student academic performance. *J Comp Electr Electron Eng Sci, 3*(2), 41-46.

Received: 08.07.2025 • **Accepted**: 27.08.2025 • **Published**: 03.10.2025

ABSTRACT

In this study, various models were developed using machine learning algorithms to predict the academic performance of students in the Programming Course at Kırıkkale University. The data, collected through a survey from 170 associate degree students, consists of 9 input attributes, including demographic information, high school education information, academic status, and socio-economic characteristics, and one output attribute. The data were analysed using the WEKA software. According to the results, the J48 algorithm was identified as the most successful model with an accuracy rate of 96.11%. The Random Forest and k-NN algorithms also closely followed J48 with accuracy rates of 95.83%, demonstrating high predictive performance. The Naive Bayes algorithm demonstrated the lowest classification accuracy with an accuracy rate of 78.06%. In terms of error rates, the Random Forest algorithm showed the best performance with the lowest MAE, RMSE, RAE, and RRSE values. This study demonstrates the applicability of machine learning techniques in education and proves that they can be used as a tool for early detection of at-risk students. In future studies, it is planned to improve the performance of the models and test their generalizability by using larger datasets and different machine learning algorithms.

Keywords: Machine learning, educational data mining, academic performance prediction, WEKA

INTRODUCTION

Predicting student success in the education system offers significant advantages to both students and teachers. Measuring and understanding the effects of numerous variables on student performance is important for enhancing academic success. Today, the biggest problem for students graduating from higher education is the problem of not being able to find a job. In addition, the fact that the education and competencies they receive do not match the personnel competencies in the sector is another important problem. The aim of educational institutions is to ensure that students receive appropriate education, competencies and skills in order to eliminate these problems. In this way, the student can be competitive in the sector and the labor market. It is important to determine in advance whether the student will be successful in this regard. In this process, the student's education can be intervened in and his/her success can be ensured. Thanks to the early determination of performance, the performance of a student who may be unsuccessful can be increased, and he/she can be directed to different skills and competencies thanks to educational coaches. Additional training can be provided. Moreover, this success will not only be the success of the student but also of the educational institution. Analyzing students' academic progress throughout their educational journey provides university administrators with valuable insights into each student's probability of success. In conventional practice, instructors evaluate this by monitoring classroom interactions and recognizing students who exhibit a higher risk of dropout, thereby enabling early intervention.

However, in actualy educational settings, the frequency and quality of interactions between instructors and students are gradually decreasing. This trend is largely due to the increasing number of working students and the growing accessibility of online learning materials. As a result, it has become more challenging to detect at-risk students through conventional methods.

To improve the accuracy of student success predictions, data mining and machine learning techniques offer effective solutions. In today's digital age, educational institutions collect large and diverse datasets. However, much of this data often remains underutilized. Leveraging these valuable resources requires advanced analytical tools, among which machine learning algorithms (MLAs) stand out as particularly powerful.

The goal of this study is to identify the specified variables affecting student success and to choose the most effective machine learning algorithm to predict this success. For this purpose, data obtained from a survey conducted on 170 students studying at Kırıkkale University were used. The survey questions constitute the input attributes. The dataset comprises three main feature categories: individual attributes, educational context, and socio-economic variables. Using the

Corresponding Author: Mahmut Ünver, munver@kku.edu.tr

CfsSubsetEval attribute evaluator and the GreedyStepwise search method in the WEKA software, the 9 most important attributes affecting student success were selected (mathematics course grade, graduated high school type, city of origin, parental education status, preparatory course, income status, gender, residence, and working in an additional job status).

The most fundamental course for computer programming students is the programming course. Success in the programming course is an indicator of the student's academic success. Therefore, the programming course was selected for performance estimation.

In the study, models were developed using the J48 decision tree, Naive Bayes, logistic regression, K-nearest neighbors (k-NN), and Random Forest algorithms, which are known to give successful results in the literature, to predict the programming course grade (output attribute). The performances of the models were compared using TP (true positive) rate, precision, F-measure, accuracy, and error criteria [MAE (mean absolute error), RMSE (root mean square error), RAE (relative absolute error), and RRSE (root relative squared error)]. This study is unique in terms of comparing the performance of different machine learning algorithms in predicting the academic success of Kırıkkale University students and determining the most suitable algorithm.

The following sections of the article include a literature review, materials and methods, the performance of the developed models, comparison of the results, and conclusion sections. A flowchart showing the general steps of the manuscript is shown in **Figure 1**.

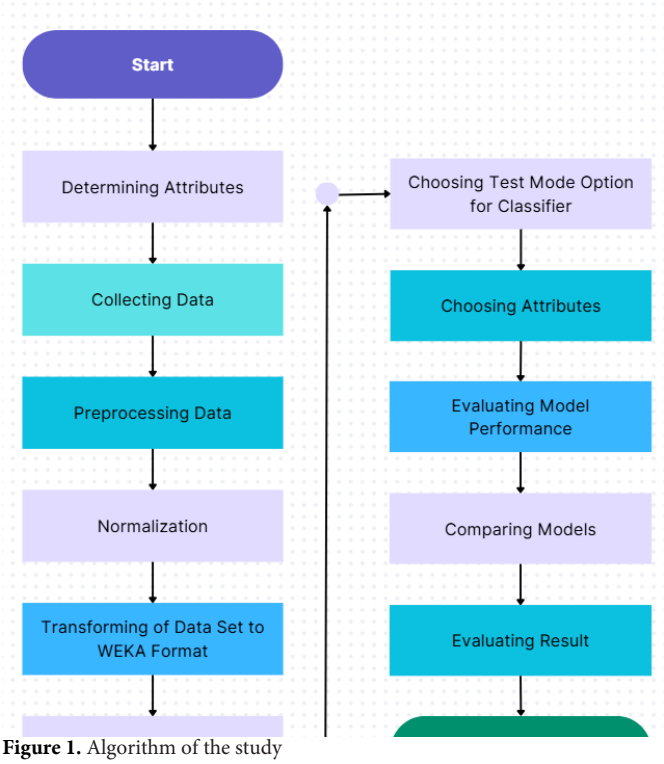

Figure 1. Algorithm of the study

LITERATURE REVIEW

In recent years, machine learning techniques have started to be widely used in the field of education. Algorithms such as support vector machines (SVM), decision trees (DT), k-nearest neighbors (k-NN), and artificial neural networks (ANN) are generally preferred in student performance prediction studies.

Kotsiantis et al. (2004) developed algorithms to predict student success in distance education. In the study, where Naive Bayes, decision trees, and other methods were compared, it was shown that the Naive Bayes method gave better results in many scenarios.

Rastrollo-Guerrero et al. (2020) evaluated machine learning techniques used to analyze and predict student performance. In this study, where supervised learning algorithms were compared, it was stated that SVM in particular gave the best results.

Hasan et al., (2020) stated that the success rate of the models increased when multiple data sources were used.

Aghalarova and Bozkurt Keser (2021) aimed to predict the academic achievements of secondary school students in mathematics and Portuguese courses with the artificial neural network algorithm. In this study, the imbalanced data problem was solved with the SMOTE algorithm, and hyperparameter optimization was performed with the random search method. With the multilayer perceptron (MLP) algorithm, 97.0% accuracy was achieved in mathematics and 92.3% in Portuguese.

In the study conducted by Chen and Zhai (2023), the performance of machine learning methods was investigated. To enable a comparative analysis of various methods, four evaluation metrics along with visual representations are introduced. The experimental findings indicate that the random forest algorithm demonstrates consistently strong generalization across all selected datasets.

Pinto and Paquette (2024) examined the use of deep learning techniques in the field of educational data science. In the literature review on the applications of deep artificial neural networks in education, it was found that deep learning algorithms are effective in areas such as detecting student behaviors and analyzing open-ended student responses.

Xiao et al. (2023) evaluated the applications of large language models in education. The uses of technologies such as deep learning, pre-training, and reinforcement learning in education were examined, and it was stated that large language models have the potential to increase teaching quality and transform educational models.

In the study by Nahar et al. (2021), a dataset containing data from 395 students from a university's undergraduate program was used. Using attributes such as students' demographic information, academic records, and behavioral characteristics, models were developed with classification algorithms such as decision table, Naive Bayes, random forest, OneR, JRip, k-NN, and multilayer perceptron in WEKA software. The performances of the models were compared based on the accuracy rate. According to the results of the study, the random forest algorithm showed the best performance with an accuracy rate of 97.5%.

In the study by Nedeva & Pehlivanova (2021), WEKA software was used. The authors used data they obtained from a survey applied to 115 engineering students. They developed classification models with BayesNet, multilayer perceptron (MLP), sequential minimal optimization (SMO), and J48 algorithms in WEKA software; The models were compared with TP rate, precision, F-measure, accuracy and error criteria (MAE, RMSE, RAE and RRSE). According to the results of

the study, the multilayer perceptron (MLP) algorithm showed the best performance with an accuracy rate of 97.39%.

In exhibited the lowest performance Abu Zohair's study (2019), they used a classifier to predict the performance of postgraduate students at The British University in Dubai in each course. The success of classifier A reached the highest accuracy rate with 79% and 77%.

METHODS

The study was conducted with the approval of the Kırıkkale University Social and Human Sciences Researches Ethics Committee (Date: 17.04.2025, Decision No: 330840). All procedures were carried out in accordance with ethical standards and the principles of the Declaration of Helsinki.

In this study, models that predict academic success were developed using machine learning algorithms in the WEKA environment, using survey data from 170 students studying at Kırıkkale University.

Data Collection and Preparation

In the data collection phase, data obtained from a survey conducted on 170 associate degree students studying at Kırıkkale University were used. The questionnaire consists of questions about students' demographic information, academic backgrounds, socio-economic status, and learning environments. The data were transferred to an Excel file and organized with the attributes in Table 1.

Table 1. Attributes and values for the data set

Type	Attributes	Values
Input	Mathematics_success_grade	AA: 5, BA: 4, BB: 3, CA: 2, CB: 1, Fail: 0
Input	Preparatory_course	Yes: 1, No: 0
Input	Graduated_high_school_type	Other_vocational_program: 1, IT_vocational_program: 2, AL:3, SL:4, FL:5
Input	City_of_origin	outside_Kırıkkale: 0, Kırıkkale: 1
Input	Gender	F: 1, M: 0
Input	Parental_education_status	Graduate: 4, Undergraduate: 3, Associate: 2, High_School:1, Other:0
Input	Income_status	0-20000:1, 20001-30000:2, 30001-50000:3, >50001:4
Input	Residence	Kırıkkale: 1, commuter: 2
Input	Working_additional_job	Yes: 1, No: 0
Output	Programming_algorithm_success_grade	AA: 5, BA: 4, BB: 3, CA: 2, CB: 1, Fail: 0

The values of the attributes were normalized, and the dataset was converted into a WEKA-compatible ARFF file.

Data Preprocessing

At this stage, the collected data were prepared for analysis. Missing data were removed from the dataset. The data preprocessing stage comprises multiple steps, which can be illustrated as follows: data collection, data cleaning, data transformation, and data normalization.

Data collection: Data obtained from the survey results were collected in Microsoft Excel format.

Data cleaning: Surveys with missing data were removed from the dataset.

Data transformation: Data in the Excel file were converted into ARFF (attribute-relation file format) format, which can be used by WEKA, using Python.

```
data = pd.read_excel('anket_verileri.xlsx')

arff_data:
  'description': 'Ogrenci Basari Tahmini',
  'relation': 'ogrenci_basari',
  'attributes':
    ['Mathematics_Success_Grade', ('5', '4', '3', '2', '1', '0')],
    ['Preparatory_Course ', ('1', '0')],
    ['Graduated_High_School_Type ', ('1', '2', '3', '4', '5')],
    ['City_of_Origin', ('0', '1')],
    ['Gender', ('1', '0')],
    ['Parental_Education_Status', ('4', '3', '2', '1', '0')],
    ['Income_Status', ('1', '2', '3', '4')],
    ['Residence', ('1', '2')],
    ['Working_Additional_Job', ('1', '0')],
    ['Programming_Algorithm_Success_Grade', ('5', '4', '3', '2', '1', '0')]
  'data': data.values.tolist()
with open ('ogrenci_basari.arff', 'w') as f:
  arff.dump(arff_data, f)
```

Data normalization: Attributes with numerical values were normalized in the [0,1] range. This process was performed using the "Filters -> Unsupervised -> Attribute -> Normalize" filter via the WEKA interface.

Attribute Selection

To achieve the research objective, attribute selection was performed before classification. The 9 most important attributes were determined using attribute selection algorithms in WEKA. The "CfsSubsetEval" attribute evaluator and the "GreedyStepwise" search method were used for this process. The attribute selection output is shown in Table 2.

Table 2. Attribute selection output

Attributes	Selected folds count	Order of importance
Mathematics_success_grade	10	1
Preparatory_course	0	9
Graduated_high_school_type	2	5
City_of_origin	10	1
Gender	1	7
Parental_education_status	1	7
Income_status	2	5
Residence	10	1
Working_additional_job	4	4

Classification Algorithms

In this study, five different machine learning algorithms, which are known to give good results in student success prediction in the literature (Batool et al., 2023) and are included in the WEKA software, were selected to predict the students' Programming_Algorithm (output attribute). These algorithms are:

J48 (decision trees): A classification algorithm that creates easy-to-interpret and understandable decision trees based on the C4.5 algorithm, which is an improved version of the ID3 algorithm.

Naive Bayes: A probabilistic classification algorithm based on Bayes' Theorem that assumes that the attributes are independent of each other (naive assumption).

Logistic regression: A statistical method that uses the logistic function to model probabilities when the dependent variable is categorical.

K-nearest neighbors (kNN): An instance-based learning algorithm that determines the class of a data point by looking at the class of its nearest "k" neighbors and using majority voting.

Random forest: A powerful and flexible ensemble learning algorithm that works by training multiple decision trees and combining their predictions.

These selected algorithms were applied through the WEKA software using nine input attributes (Mathematics_Success_Grade, Preparatory_Course, Graduated_High_School_Type, City_of_Origin, Gender, Parental_Education_Status, Income_Status, Residence, Working_Additional_Job) and one output attribute (Programming_Algorithm_Success_Grade).

Performance Evaluation Metrics

To evaluate the performance of each classification, the confusion matrix obtained for each classification was used. The classification values for student performance are as follows; AA, BA, BB, CA, CB, and Fail. Therefore, each confusion matrix is created according to 25 classification results. From the confusion matrix, the following parameters can be read for each class; true positive (TP) prediction, true negative (TN) prediction, false positive (FP) prediction, and false negative (FN) prediction.

The parameters used to compare the algorithms are calculated as follows (Sokolova & Lapalme, 2009), (Powers, 2020), (Tharwat, 2021):

True Positive Rate (TP Rate)

It is the ratio of correctly assigned instances to a class.

$$TP\ rate=TP/(TP+FN) \quad (1)$$

Precision (P)

This metric represents the proportion of correctly identified positive cases relative to all instances predicted as positive.

$$P=TP/(TP+FP) \quad (2)$$

F-measure (FM)

It is calculated as the harmonic mean of the Precision (P) and Recall (R) metrics.

$$Fm=2\ PR/(P+R) \quad (3)$$

Here, R=TP/(TP+FN).

Accuracy

This metric indicates the proportion of correctly classified instances out of the total number of cases.

$$Accuracy=(TP+TN)/(TP+TN+FP+FN) \quad (4)$$

The various error metrics employed in classification methods are presented below.

Mean Absolute Error (MAE)

It is an estimate of how different the predictions are from the values.

$$MAE=(1/n)*\Sigma\ |pi-ai| \quad (5)$$

Here, n is the number of errors, and |pi - ai| are the absolute errors.

Root Mean Square Error (RMSE)

It quantifies the discrepancy between the forecasted values and the empirically observed outcomes.

$$RMSE=\sqrt{[(\Sigma\ (pi-ai)\hat{}2)/n]} \quad (6)$$

pi are the predicted values, ai are values at time/place i (Nedeva & Pehlivanova, 2021).

Relative Absolute Error (RAE)

RAE indicates the ratio of the errors made by the model to the errors that would be made by a simple mean predictor (i.e., a model that predicts the mean value of the output variable for all instances). This metric allows us to understand how much better (or worse) the model's predictions are than a naive mean prediction. A low RAE value means better model performance.

$$RAE=(\Sigma\ |pi-ai|)/(\Sigma\ |ai-\bar{a}|) \quad (7)$$

Where:

- **pi:** The predicted value of the i-th instance by the model
- **ai:** The actual value of the i-th instance
- **ā:** The mean of the actual values of all instances
- **n:** The total number of instances

Root Relative Squared Error (RRSE)

The RRSE is the square root of the ratio of the sum of the squares of the errors made by the model to the sum of the squares of the errors that would be made by a simple mean predictor. Like RAE, RRSE allows us to understand how much better (or worse) the model's predictions are than a naive mean prediction. A low RRSE value means better model performance.

$$RRSE=\sqrt{[(\Sigma\ (pi-ai)^2)/(\Sigma\ (ai-\bar{a})^2)]} \quad (8)$$

Where:

- **pi:** The predicted value of the i-th instance by the model
- **ai:** The actual value of the i-th instance
- **ā:** The mean of the actual values of all instances
- **n:** The total number of instances

For each algorithm, the default parameter settings of WEKA were rearranged.

- For the J48 algorithm, confidenceFactor=0.25 and minNumObj=2 were used.
- For the Naive Bayes algorithm, useKernelEstimator=False and useSupervisedDiscretization=False values were used.
- For the Logistic Regression algorithm, ridge= 1.0E-8 and maxIts=-1 values were used.

- For the k-NN model, K=1, distance weighting=No distance weighting, and nearestNeighbourSearchAlgorithm= euclideanDistance were selected.
- For the random forest algorithm, numIterations=100, maxDepth=0, and numFeatures=0 values were used.

The models used were trained separately with 10-fold Cross-validation and Percentage split (70%) options.

DEVELOPED MODELS AND THEIR PERFORMANCES

While developing the models, after loading the dataset, the classifier was first selected. Then the class attribute was selected, and the parameters of the classifier were adjusted to give the most accurate result. Each of the models was first developed with cross-validation (10 folds) Test options, and then with percentage split (70%). All 5 models gave more accurate results with percentage split (70%). The correctly classified instances values of the models according to the used test option are shown in Table 3.

Table 3. Correctly classified instances of model performances

Algorithm	Cross-validation (10 folds) (%)	Percentage split (70%) (%)
J48	94.8333	96.1111
Naive Bayes	77.3333	78.0556
Logistic regression	80.2500	80.5556
k-NN (k=1)	94.4167	95.8333
Random forest	94.4167	95.8333

The comparison of the weighted average in the detailed accuracy by class and the performance values of the obtained models are shown in Table 4.

According to the results presented in Table 3 and Table 4, the performances of the five different machine learning models developed were compared using TP rate, precision, F-measure, accuracy, MAE, RMSE, RAE, and RRSE metrics. The J48 algorithm showed the highest success with an accuracy rate of 96.1111%, a TP rate of 0.961, a Precision of 0.963, and an F-measure of 0.968. In addition, the random forest algorithm had the lowest error values (MAE: 0.012, RMSE: 0.0786, RAE: 4.3656%, RRSE: 21.1193%).

It turned out that the k-NN and Random Forest algorithms have the lowest TP Rate (0.958). In addition, the Correctly values of these algorithms (95.8333) produced the closest values to the J-48 algorithm.

The Naive Bayes algorithm had the lowest values with a correctly rate of 78.0556, a TP Rate of 0.781, a precision of 0.782, an F-measure of 0.778, a MAE of 0.111, a RMSE of 0.2394, a RAE percentage of 40.3395, and a RRSE percentage of 64.3366.

In summary, according to the results, the J48 (decision trees) algorithm stands out as the most successful model in predicting student success, while the Naive Bayes algorithm gave the lowest performance values.

The comparison of the developed algorithms according to their accuracy rates is shown in Figure 2.

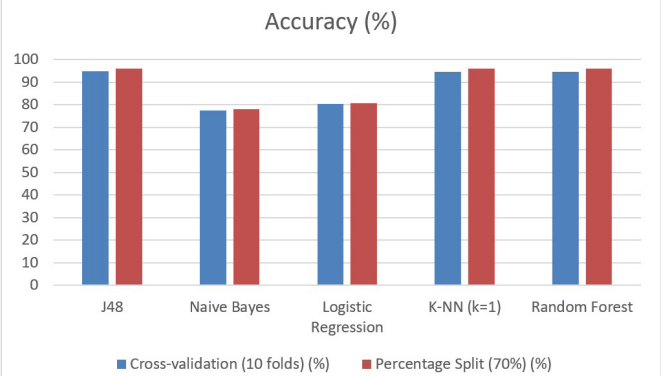

Figure 2. Accuracy rates (%) for algorithms

MAE (mean absolute error) and RMSE (root mean squared error) are error metrics used to evaluate the performance of regression and classification models, and it is desired for these values to be low for an accurate model. The MAE and RMSE values of the developed models are shown in Figure 3. According to the MAE value, the random forest, k-NN, and J48 models produced the lowest values, from smallest to largest. According to the RMSE value, the Random Forest, k-NN, and J48 models produced the lowest values.

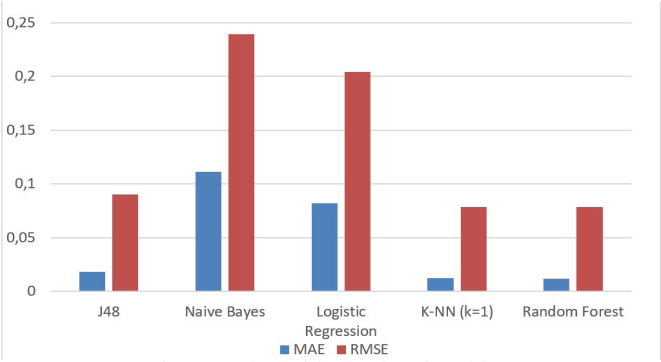

Figure 3. MAE and RMSE values of the developed models

RESULTS

In this study, models were developed using different machine learning algorithms to predict the success of Kırıkkale University students in the programming course. The survey data collected from 170 students consist of 9 input attributes, including demographic information, high school education information, academic status, and socio-economic characteristics, and the output attribute labeled as "programming course grade" (AA, BA, BB, CA, CB, Fail). After the data preprocessing steps, attribute selection was made in the WEKA software, and the 9 attributes with the

Table 4. Error and performance values of models

	TP rate	Precision	F-measure	MAE	RMSE	RAE (%)	RRSE (%)
J48	0.961	0.963	0.968	0.018	0.0901	6.5396	24.2241
Naive Bayes	0.781	0.782	0.778	0.111	0.2394	40.3395	64.3366
Logistic regression	0.806	0.823	0.805	0.082	0.2046	29.7822	54.9894
k-NN (k=1)	0.958	0.964	0.959	0.0121	0.0786	4.3912	21.118
Random forest	0.958	0.964	0.959	0.012	0.0786	4.3656	21.1193

highest impact were determined (mathematics course grade, graduated high school type, city of origin, parental education status, preparatory course, income status, gender, residence, and working in an additional job status).

In the study, five different machine learning algorithms (J48, Naive Bayes, logistic regression, k-NN, and random forest) known to give good results in student success prediction in the literature were used. The models were trained with 10-fold cross-validation and 70% to 30% train-test dataset separation, and their performances were evaluated using accuracy, TP rate, precision, F-measure, MAE, RMSE, RAE, and RRSE metrics.

According to the results obtained, the J48 algorithm was the most successful model with an accuracy rate of 96.11%. This model was followed by the random forest and k-NN algorithms with an accuracy rate of 95.83%. The Naive Bayes algorithm showed the lowest performance with an accuracy rate of 78.06%. When evaluated in terms of error rates, the random forest algorithm produced the lowest MAE (0.012), RMSE (0.0786), RAE (4.3656%), and RRSE (21.1193%) values, making it the model with the lowest error rate. The J48 algorithm also drew attention with its low error rates (MAE: 0.018, RMSE: 0.0901, RAE: 6.5396%, RRSE: 24.2241%).

In this study, it was observed that machine learning algorithms can achieve high accuracy rates in predicting student success. The results of our study show that the J48 and random forest algorithms have high potential in predicting the programming course success of Kırıkkale University students. These models can be used to identify at-risk students, especially those who need to be accurately predicted. In addition, in light of the findings obtained, it can be said that the mathematics course grade, city of origin, and residence attributes have a significant impact on student success.

In future studies, more advanced models can be created by using more comprehensive questionnaires and adding additional attributes such as student absenteeism information and study habits. In addition, the performances of the models can be compared by trying different attribute selection methods and different machine learning algorithms. Finally, the models used in this study were developed to predict the programming course success of Kırıkkale University students. Conducting similar studies in different universities and different courses is important in terms of the generalizability of the results obtained.

Limitations

In this study, it was observed that machine learning algorithms can achieve high accuracy rates in predicting student success. There are also some limitations to this study. First, the dataset is limited to 170 students. With a larger dataset, it is possible to increase the performance of the models and obtain more reliable results. Second, the questionnaire questions used may not cover all the factors that may affect students' academic success.

CONCLUSION

Consequently, this study shows that machine learning algorithms can be used as an effective tool in predicting student success and can help educators in identifying at-risk students and taking necessary measures.

ETHICAL DECLARATIONS

Ethics Committee Approval

The study was carried out with the permission of the Kırıkkale University Social and Human Sciences Researches Ethics Committee (Date: 17.04.2025, Decision No: 330840).

Informed Consent

All students signed and free and informed consent form.

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Abu Zohair, L.M. (2019). Prediction of Student's performance by modelling small dataset size. *Int J Educ Technol High Educ, 16*(1), 1-18. doi:10.1186/s41239-019-0160-3

Aghalarova, S., & Keser, S. B. (2021). Önerilen yapay sinir ağı algoritması ile ortaokul öğrencilerin akademik performansının tahmini. *Veri Bilimi*, *4*(2), 19-32.

Batool, S., Rashid, J., Nisar, M. W., Kim, J., Kwon, H. Y., & Hussain, A. (2023). Educational data mining to predict students' academic performance: a survey study. *Edu Informat Technol*, *28*(1), 905-971. doi: 10.1007/s10639-022-11152-y

Chen, Y., & Zhai, L. (2023). A comparative study on student performance prediction using machine learning. *Edu Informat Technol*, *28*(9), 12039-12057. doi:10.1007/s10639-023-11672-1

Hasan, R., Palaniappan, S., Mahmood, S., Sarker, K. U., & Abbas, A. (2020). Modelling and predicting student's academic performance using classification data mining techniques. *Int J Business Informat Systems*, *34*(3), 403-422. doi:10.1504/IJBIS.2020.108649

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Predicting students' performance in distance learning using machine learning techniques. *Appl Artific Intell*, *18*(5), 411-426. doi:10.1080/08839510490442058

Nahar, K., Shova, B. I., Ria, T., Rashid, H. B., & Islam, A. H. M. S. (2021). Mining educational data to predict students performance. *Edu Informat Technol*, *26*(5), 6051-6067. doi:10.1007/s10639-021-10575-3

Nedeva, V., & Pehlivanova, T. (2021). Students' performance analyses using machine learning algorithms in WEKA. IOP Conference Series: Materials Science and Engineering, *1031*(1), 012061. doi:10.1088/1757-899X/1031/1/012061

Pinto, J. D., & Paquette, L. (2024). Deep Learning for Educational Data Science. In Trust and Inclusion in AI-Mediated Education: Where Human Learning Meets Learning Machines (pp. 111-139). Springer Nature Switzerland. doi:10.1007/978-3-031-64487-0_6

Powers, D. M. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*. doi:10.48550/arXiv.2010.16061

Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students' performance by means of machine learning: a review. *Appl Sci*, *10*(3), 3. doi:10.3390/app10031042

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Informat Process Manag*, *45*(4), 427-437. doi:10.1016/j.ipm.2009.03.002

Tharwat, A. (2021). Classification assessment methods. *Appl Comput Informat*, *17*(1), 168-192. doi:10.1016/j.aci.2018.08.003

Xiao, C., Xu, S. X., Zhang, K., Wang, Y., & Xia, L. (2023). Evaluating reading comprehension exercises generated by LLMs: a showcase of ChatGPT in education applications. In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023) (pp. 610-625). Association for Computational Linguistics. doi:10.18653/v1/2023.bea-1.52

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0028

Genetic algorithm-based optimization of PID controller parameters for DC motor speed control

Zafer Civelek, Mustafa Teke
Department of Electrical Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkiye

Cite this article as: Civelek, Z., & Teke, M. (2025). Genetic algorithm-based optimization of PID controller parameters for DC motor speed control. *J Comp Electr Electron Eng Sci, 3*(2), 47-53.

Received: 03.07.2025 ◆ **Accepted**: 01.09.2025 ◆ **Published**: 03.10.2025

ABSTRACT

This study investigated the optimization of PID controller parameters for DC motor speed control using genetic algorithms, comparing the results with the traditional Ziegler-Nichols tuning method. A comprehensive mathematical model of a DC motor was developed in MATLAB/Script, incorporating realistic operating constraints and dynamic load variations to simulate practical industrial applications. The genetic algorithm was configured with a crossover rate of 90%, mutation rate of 3%, population size of 20 individuals, and single-point crossover method, running for 100 iterations. The fitness function incorporated multiple performance criteria including total error, settling time, and overshoot. The GA-optimized PID controller with parameters kp=95.984, ki=0.250, and kd=13.005 outperformed the Ziegler-Nichols-tuned controller (kp=20, ki=0.5, kd=3) across all performance metrics. Most notably, the overshoot was reduced by approximately 59% (from 4.724% to 1.922%), while maintaining comparable rise time. The controller also demonstrated excellent robustness when subjected to variable load torques of 20 N·m and 10 N·m at different points during simulation.

Keywords: PID controller, genetic algorithm, DC motor, Ziegler-Nichols method, speed control optimization

INTRODUCTION

DC motors have emerged as fundamental components of contemporary industrial control systems due to their versatility, reliability characteristics, and relative simplicity of control mechanisms. These motors are utilized across a broad spectrum of applications including robotics, electric vehicles, aerospace systems, and various manufacturing processes (Ortatepe, 2023). One of the most critical aspects of DC motor control is achieving precise speed regulation necessary for optimal performance and efficiency under dynamic operating conditions. To address this challenge, various control strategies such as proportional-integral-derivative (PID) controllers have emerged as widely preferred solutions due to their robustness and implementation simplicity. The control performance of PID controllers largely depends on the accurate determination of their parameter configurations (Borase et al., 2021). The mathematical modeling of DC motors plays a critical role in the design and optimization of control strategies. A comprehensively characterized DC motor model precisely represents the dynamic behaviors of the motor, including the relationships between armature voltage, current, and rotational velocity (Allaoua et al., n.d.; Tiwari et al., 2018). Researchers have developed various mathematical models over the years, ranging from simple linear approximations to complex nonlinear formulations that incorporate significant factors such as friction, magnetic saturation, and load variations. These models not only provide the foundation for simulation and analysis of motor behavior under various operating conditions but also contribute significantly to the formulation of advanced control techniques. Despite their widespread implementation, PID controllers encounter significant limitations when applied to systems with complex dynamics and time-varying parameters.

The main problem addressed in this work is to ensure reliable tracking of DC motor speed under realistic industrial constraints (time delay, supply voltage limitations, sudden load torque changes) without violating overshoot and settling time limits; classical Ziegler-Nichols tuning often cannot guarantee target performance under these conditions. Traditional parameter tuning methodologies, such as Ziegler-Nichols, frequently prove inadequate in delivering optimal performance across variable operational scenarios (Patel, 2020). To overcome these limitations, researchers have developed various optimization algorithms to enhance PID controller performance. These approaches encompass classical optimization techniques such as gradient descent and pole placement, as well as contemporary metaheuristic methods including particle swarm optimization (PSO), genetic algorithms (GA), and artificial neural networks (ANN) (Gaing, 2004a, 2004b; Pereira & Pinto, 2005). These techniques aim to automate the parameter tuning process and adapt controller parameters to changing system conditions, thereby improving control precision and robustness (Figueiredo et al., 2023). Manuel et al. (2023) studied a DC motor speed model

Corresponding Author: Zafer Civelek, zafercivelek@karatekin.edu.tr

based on MATLAB/Simulink and used the standard motor model as material; optimized PID gains with meta-heuristic methods such as EO, PSO, TLBO, DE, and GA and compared them with FLC; as a result, they reported that optimized PIDs outperformed classical tuning, while FLC provided the lowest overshoot and settling time (Manuel et al., 2023). Huang et al. (2018) used a model that includes PWM/saturation effects as material in BLDC speed control; designed a self-tuning fuzzy PID with anti-windup block as a method and tested it under sudden load and reference jumps; as a result, it showed shorter settling time, lower overshoot, and better disturbance rejection compared to the classical PID (Jigang et al., 2018). Ahangari Sisi et al. (2023) established a materially delayed system in the arm assembly driven by a time-delayed DC motor; as a method, they compensated for the delay with the Smith estimator and adjusted the PID according to the non-delayed part; as a result, they stated that they reduced the delay-induced oscillations and obtained a faster/damped response (Ahangari Sisi et al., 2023). Idir et al. (2018) tested PID and fractional order FOPID with DC motor speed model as material; optimized ITAE criterion with DE and PSO as method; as a result, they showed that FOPID surpassed PID in terms of both tracking and noise immunity (Idir et al., 2018). Saravanan et al. (2025) took the DC motor control system as the material and adjusted the PID gains according to ITAE with Kookaburra and Red Panda optimization algorithms as the method; as a result, they reported significant improvements in rise/settlement times and robustness (Saravanan et al., n.d.). Acharya et al. (2021) worked with an armature-controlled DC motor model in MATLAB/Simulink as a material; adjusted the PID gains according to ITAE with Archimedes Optimization (AOA) and Dispersive Flies Optimization (DFO) as a method and compared them with Ziegler-Nichols and PSO; as a result, they showed that AOA/DFO reduces the rise and settling times and reduces overshoot, and DFO converges faster (Acharya et al., 2021). Sharma et al. (2022) used DC motor speed model as material and compared PSO-based PID tuning with ZN-PID on four performance indices (IAE, ISE, ITAE, ITSE). As a result, they reported that PSO-PID gives superior and better transient regime in all indices (Sharma et al., 2022). Yadav and Gupta (2024) considered a DC motor speed system with a materially delayed dynamics and designed a modified Smith estimator + PID with a direct synthesis approach and tested the servo/regulator conditions and ±30% parameter deviations. As a result, they showed that ISE/IAE is significantly reduced and the immunity to disturbances is increased compared to the classical PID (Yadav & Gupta, 2024).

This study focuses on the mathematical modeling of DC motors and the metaheuristic optimization of PID controllers for speed regulation. The research examines the formulation of mathematical models that characterize the structural parameters, fundamental operating principles, and dynamic behaviors of DC motors. The work systematically analyzes the challenges associated with PID controller parameter tuning and evaluates proposed optimization methodologies to overcome these challenges. Additionally, the impact of sudden load variations and environmental factors on controller performance is comprehensively addressed. Finally, this investigation highlights recent advancements in adaptive and intelligent control techniques that offer promising solutions for overcoming the limitations of conventional PID controllers. This study emphasizes the importance of parameter tuning of PID controllers in DC motor control. It also presents how parameter optimization makes the control system stable, accurate and efficient.

In industrial applications, DC motors are subjected to time-varying load torques, measurement and processing delays, and saturation of the control signal. Under these conditions, classical PID tuning methods such as Ziegler–Nichols may fail to achieve the desired balance between overshoot, settling time, and tracking error, and are sensitive to system parameter uncertainties. This study addresses this tuning problem, where PID gains cannot be determined in closed form and must be optimized under a multi-objective performance metric (total error, settling time, overshoot). As a solution approach, a genetic algorithm (GA)-based PID optimization is proposed using a model that includes realistic constraints such as time delay (~100 ms), supply voltage limit (35 V), and variable load torques applied at specific instants (e.g., 20 and 10 N m at 35 and 65 seconds). This approach is systematically compared with the classical Ziegler-Nichols. The contribution of this study to the literature is as follows:

- Mathematical modeling of DC motors with comprehensive characterization of dynamic behaviors
- Systematic analysis of PID controller parameter tuning challenges
- Evaluation of metaheuristic optimization methodologies for PID controllers
- Assessment of how sudden load variations and environmental factors affect controller performance
- Review of recent advancements in adaptive and intelligent control techniques

Table 1 shows a summary of the literature.

METHODS

Methodology and Material

In this study, optimized PID controller parameters for speed control of DC motor were developed. First, a comprehensive mathematical model of DC motor was created in MATLAB/Script environment and operating constraints and variable load conditions that may be encountered in industrial applications were included in the simulation. Genetic algorithm (GA) approach was used for optimization of PID controller parameters. GA was configured with 90% crossover rate, 3% mutation rate, 20 individual population size and single point crossover method and was run for 100 iterations. In the optimization process, a fitness function was used that considered multiple performance criteria including total error, settling time and overshoot values. First, a study was conducted on the DC motor model.

DC Motor Mathematical Model Expansion

The mathematical model of a DC motor comprises the integration of electrical and mechanical dynamics to fully characterize its operation. This section presents a comprehensive description encompassing the differential equations, state-space representations, transfer functions, and practical considerations employed in the mathematical modeling of DC motors (Ortatepe, 2023a). Equation 1 below the electrical behavior of the armature circuit is derived using Kirchhoff's Voltage Law (KVL):

Table 1. Comparison with the literature

Authors & year	Title	Contributions	Implementation	Performance metrics
(Khettab et al., 2025)	Performances improvement of DC motor using a fractional order adaptive PID controller...	Optimization of Fractional Order Adaptive PID (FAPID) controller using GA; better performance compared to APID	GA used to optimize FAPID parameters (Kp, Ki, Kd, λ, μ).	Rise time: 0.0084s, settling time: 0.0683s, overshoot: 0%, MAE: 0.0032 rad/s
(M. V Patel & Pathak.,2015)	PID tuning using genetic algorithm for DC motor positional control system	GA-optimized PID outperforms Ziegler-Nichols with lower overshoot and faster response time	GA used ITAE, IAE, and ISE objective functions to optimize PID parameters.	Rise time: 0.06s (GA-IAE), settling time: 0.741s, overshoot: 38.7%
(Tiwari et al., 2018b)	Control of DC motor using genetic algorithm based PID controller	GA-optimized PID shows lower overshoot (0.408%) and faster rise time compared to Ziegler-Nichols	MSE and ITAE objective functions used; GA parameter ranges [0-700, 0-8000, 0-5]	Rise time: 0.0202s, settling time: 0.917s, overshoot: 0.408%
(Mahmood et al., 2023)	Design of fractional order PID controller based on genetic algorithm optimization...	GA-optimized FOPID for VTOL system; comparison of ISE, ITSE, and MSE objective functions	GA optimized FOPID parameters (Kp, Ki, Kd, λ, μ); population size 40	Rise time: 0.00259s, settling time: 0.0044s, overshoot: 0.127%
(Jamil & Moghavvemi, 2021)	Optimization of PID controller tuning method using evolutionary algorithms	Comparison of GA and PSO for PID tuning; PSO outperforms GA in convergence speed	GA and PSO used to optimize PID parameters; GA parameters: population=35, iterations=40	Rise time: 0.0048s, settling time: 0.0079s, overshoot: 0%
(Mahfoud et al., 2021)	A new strategy-based PID controller optimized by genetic algorithm for DTC of the doubly fed induction motor	GA-optimized PID for DTC of DFIM; improved speed, torque, and flux ripples	GA optimizes PID parameters (Kp, Ki, Kd) using weighted ISE, IAE, and ITAE	Response time:18.2ms, overshoot: 0%, torque ripples:2.05Nm, THD: 4.8%
(Yan & Zhou, 2020)	Application to optimal control of brushless DC motor with ADRC based on genetic algorithm	Proposed ADRC optimized by GA for BLDCM. Introduced ISE-ITAE fitness function to balance dynamic and static performance. Demonstrated superiority over PID control	Adaptive GA: Variable crossover (Pc*Pc*) and mutation (Pm*Pm*) probabilities. Fitness function: ISE+ITAE. Population size, coding digits, and evolution steps configured	Overshoot: 3.5% reduce, lower torque ripple, improved stability
(Suseno & Ma'Arif, 2021)	Tuning of PID controller parameters with genetic algorithm method on DC motor	Applied GA for PID tuning on DC motors. Compared GA with trial-and-error methods	Population size: 50 Generations: 100 Mutation/crossover effects: Tested varying rates	Overshoot: 2%, Settling time: 13.5s. Rise time: 2.7872s.
(Ortatepe, 2023b)	Genetic algorithm based PID tuning software design and implementation for a DC motor control system	Developed MATLAB-based GA-PID tuning software. Compared GA with conventional methods. Conducted sensitivity analysis	Fitness function: ITAE Selection: Tournament method Crossover: One-point strategy Mutation rate: 0.01 Population size: 50 Generations: 40	Reduced overshoot (e.g., 0% vs. 5% in conventional PID), Minimized steady-state error, Faster settling time (e.g., 0.4s), Robustness to parameter variations.
(Gonzales & Manzano, 2020)	Genetics algorithms based PID tuning using elastic net as model structure in non-parametric system identification	Integrated elastic net (non-parametric) system identification with GA-based PID tuning	GA parameters: 33 bits/ chromosome, mutation factor: 0.7, crossover factor: 0.3, 40 generations, population size: 20	PID GA&EN: Overshoot: 103.69%, peak time: 4.11s, Settling time: 4.79s. PID GA&TF: Settling time: 5.94s
(Ibrahim et al., 2019)	Optimal PID controller of a brushless DC motor using genetic algorithm	- Optimized PID parameters for BLDC motor using GA with ISE/ IAE criteria - Demonstrated superiority over trial-and-error and MATLAB Tuner	GA parameters: Roulette, 40 generations, population size: 20 Objective functions: ISE and IAE. Mutation: 2%, Crossover rate: 90%	Overshoot: 0 Rise time: 4.56×10-84.56×10-8s Settling time: 8.12×10-88.12×10-8s Steady-state error: 0

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad (1)$$

where:

- **Va(t):** Applied armature voltage (V)
- **Ra:** Armature resistance (Ω)
- **La:** Armature inductance (H)
- **ia(t):** Armature current (A)
- **eb(t):** Back electromotive force (EMF) (V)

Equation 2 below the back EMF is proportional to the motor's angular velocity [ω(t)]:

$$eb(t)=Ke\omega(t) \quad (2)$$

where:

Ke: Back EMF constant (V·s/rad).

Equation 3 below the mechanical subsystem follows Newton's second law for rotation:

$$J\frac{d\omega(t)}{dt} + B\omega(t) = \tau_m(t) - \tau_L(t) \quad (3)$$

where:

- **J:** Moment of inertia (kg·m²)
- **B:** Viscous friction coefficient (N·m·s/rad)
- **τm(t):** Motor torque (N·m)
- **τL(t):** Load torque (N·m)

Equation 4 below the motor torque is proportional to armature current:

$$\tau m(t)=Kt\ \ ia(t) \quad (4)$$

where:

- **Kt:** Torque constant (N·m/A). In SI units, *Kt=Ke*.

Equations 5 and 6 below define the state variables as angular velocity (*ω*) and armature current (*ia*):

$$\frac{d\omega}{dt} = -\frac{B}{J}\omega + \frac{K_t}{J} i_a - \frac{\tau_L}{J} \quad (5)$$

$$\frac{di_a}{dt} = -\frac{K_e}{L_a}\omega - \frac{R_a}{L_a} i_a + \frac{V_a}{L_a} \quad (6)$$

This linearized model is foundational for control system design.

Assuming no load torque (τL=0) and applying Laplace transforms at equation 7 below:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_t}{(L_a s + R_a)(Js + B) + K_e K_t} \quad (7)$$

Equation 8 below for simplicity if La=0(common in small DC motors):

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_t}{(R_a J s) + (R_a B + K_e K_t)} \quad (8)$$

This first-order model is often used for speed control.

In the light of these equations, the DC motor control block diagram is shown in **Figure 1**.

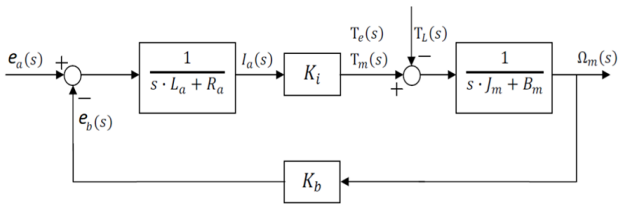

Figure 1. The DC motor control block diagram

PID Controller

The main working principle of the PID (proportional-integral-derivative) controller is based on errors. The principle is to create the difference between the desired value and the actual value, and to produce the required output value of the system by correcting this difference value with proportional, integral, and derivative components (Chauhan et al., 2023). While the proportional component provides a quick response by scaling the current output value with a coefficient when an error occurs, it cannot eliminate the steady-state error. There must be an error with the controller to produce current control output. The magnitude of the proportional coefficient affects the speed of the system's response, meaning a low coefficient can be considered as a low response and a high coefficient as a high response. However, excessively high values cause undesirable overshooting and endless oscillations in the system. The integral component is used to correct the steady-state error and to optimize the integral characteristics of the system. The integral effect of the system is inversely proportional to the time constant and is ready to intervene if the deviation is not corrected. However, a very strong integral effect can cause delays in responses to external system effects. The derivative component detects the changing trend of the error sensitively, intervenes in the system in advance, and comes into play as a correction signal. In this respect, it increases the response speed of the system to external inputs (Zhang et al., 2009). The power of the derivative component not only measures deviations or errors but also predicts changes in deviation, enabling control measures to be applied in advance. This reduces system overshooting, prevents oscillations, and shortens response time. However, if the derivative time constant is too large, the system may become unstable. The general equation of PID controller is given in equation 9.

$$u(t) = Kpe(t) + Ki \int e(t)dt + Kd\frac{de(t)}{dt} \quad (9)$$

Where:

- **u(t):** Control signal (output)
- **e(t):** Error signal (desired value - actual value)
- **Kp:** Proportional gain coefficient
- **Ki:** Integral gain coefficient
- **Kd:** Derivative gain coefficient
- **∫e(t)dt:** Integral of error with respect to time
- **de(t)/dt:** Derivative of error with respect to time

As individual components:

- **Proportional term:** P = Kp·e(t)
- **Integral term:** I = Ki·∫e(t)dt
- **Derivative term:** D = Kd·(de(t)/dt)

Ziegler-Nichols

PID controllers continue to be the predominant control solution for DC motor speed regulation due to their effectiveness, simplicity, and straightforward implementation. However, the performance of a PID controller is largely dependent on appropriate parameter tuning. Among various tuning methodologies, the Ziegler-Nichols method offers a practical approach particularly suitable for DC motor applications, as it can be implemented without detailed knowledge of motor parameters that may vary with operating conditions or aging. By presenting a systematic procedure that accommodates the variability and nonlinearities inherent in electromechanical systems, it remains a robust approach for tuning PID controllers in DC motor applications (Chauhan et al., 2023; Taguchi & Araki, 2000). Standard Ziegler-Nichols's parameters typically require modification for DC motor applications, often reducing Kp to achieve acceptable overshoot. Anti-windup mechanisms are necessary for DC motor control due to supply voltage limitations and inertial effects. In this study, PID parameters were manually obtained using the Ziegler-Nichols method. The parameter acquisition process begins by disabling the integral and derivative actions of the controller. After setting ki and kd values to zero, the proportional gain coefficient kp is increased until sustained oscillations occur. The period (Tu) of the oscillatory signal obtained at the system output is recorded. Under the saturation and delay effects in the system (to satisfy the overshoot and settling time constraints), we gradually weakened the integral effect by applying the standard "ZN initialization + fine-tuning" step; at the end of this process, the final coefficients were obtained. Finally, PID parameters are established using the table recommended by Ziegler-Nichols with the recorded Tu and the final kp value, ku, which produced the oscillation (Patel, 2020). DC motor application values of Ziegler-Nichols parameters are given in **Table 2**.

Table 2. Ziegler-Nichols DC motor parameters

Controller type	Kp	Ki	Kd
P	0.5 Ku	0	0
PI	0.45 Ku	0.54 Ku/Tu	0

Genetic Algorithm (GA)

GAs are robust metaheuristic search techniques inspired by the principles of natural selection, employed to address complex optimization problems. Initially proposed by John Holland in the 1960s and popularized by David E. Goldberg's seminal work, Genetic Algorithms in Search, Optimization, and Machine Learning in 1989, GAs represent one of the most

widely utilized forms of evolutionary computation techniques (Goldberg & Deb, 1991). These algorithms aim to solve intricate optimization problems by emulating the biological evolutionary process, which is grounded in Darwin's principles of natural selection and "survival of the fittest." Genetic algorithms offer significant advantages in tackling high-dimensional, nonlinear, and analytically challenging problems (Sivanandam & Deepa, 2008).

A genetic algorithm comprises several key components, including encoding, fitness function, selection, crossover, and mutation. In contemporary applications, various encoding strategies are employed depending on the nature of the problem. Among the most frequently used encoding methods are binary encoding, permutation encoding, value encoding, and tree encoding. Demonstrated that the choice of encoding must align with the problem's characteristics, as inappropriate encoding can significantly degrade the algorithm's performance (Rothlauf, 2003).Genetic algorithm flow chart is shown **Figure 2**.

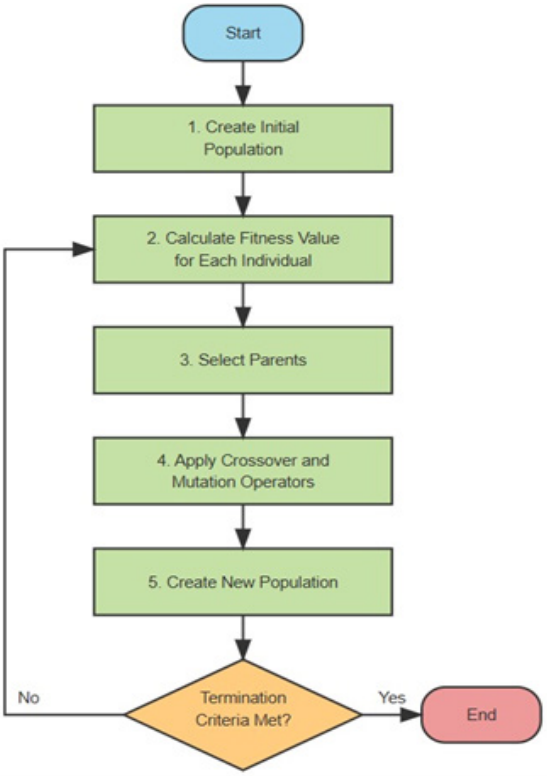

Figure 2. Genetic algorithm flow chart

The fitness function is a mathematical expression that evaluates how close everyone is to solving the problem. The researcher emphasized that the correct design of the fitness function is one of the most critical factors in the success of genetic algorithms. The selection mechanism determines which individuals in the population will have the opportunity to reproduce. Among the most used selection methods in the literature are roulette wheel selection, tournament selection, rank-based selection, and elitism (Jayachitra & Vinodha, 2014).

Crossover is the process of generating new solutions (offspring) from two parent solutions. Researchers investigated the critical role of crossover in the performance of genetic algorithms and explored the effects of different crossover operators. The primary crossover methods used in the literature include single-point crossover, multi-point crossover, uniform crossover and arithmetic crossover (Jaen-Cuellar et al., 2013; Pereira & Pinto, 2005).

Mutation is the process of introducing random changes into the solution set to increase genetic diversity and prevent the solution set from becoming trapped in local optima. Demonstrated that both excessively low and excessively high mutation rates can negatively impact the performance of the algorithm (Zahir et al., 2018; Zhang et al., 2009).

The researcher examined the applications of genetic algorithms in the design of control systems and demonstrated the effectiveness of genetic algorithms in optimizing PID controller parameters (Chauhan et al., 2023; Kasilingam, 2014; Zahir et al., 2018). In this study, the crossover rate of the genetic algorithm was selected as 90%, with a mutation rate of 3%. The population size was determined to be 20 individuals, and a single-point crossover method was implemented. The genetic algorithm was executed for 100 iterations.

RESULTS

The DC motor was modeled in the MATLAB/Script environment, and its angular velocity was controlled using a PID controller. The MATLAB/Script codes for the modeled DC motor are provided in Appendix-1. To make the study more realistic, a time delay of 100 ms was added to the DC motor. Additionally, a more realistic working environment was provided by limiting the DC source voltage to 35 volts. Load torques of 20 N.m at the 35th second and 10 N.m at the 65th second of the simulation were applied to the DC motor. The parameters of the DC motor are presented in **Table 3**.

Table 3. DC motor parameters

r = 0.35; % armature resistance
L = 0.0015; % nominal inductance
j = 0.06; % moment of inertia
b = 0.1; % damping constant
kt = 0.5; % torque constant
ke = 0.5; % voltage constant

The simulation step size was selected as 0.001 and the simulation duration as 100 ms. The PID controller coefficients kp, ki, and kd were optimized using a genetic algorithm. The coefficients obtained because of the optimization are: kp=95.984, ki=0.250, and kd=13.005. The change in the fitness function during the optimization process is shown in **Figure 3**.

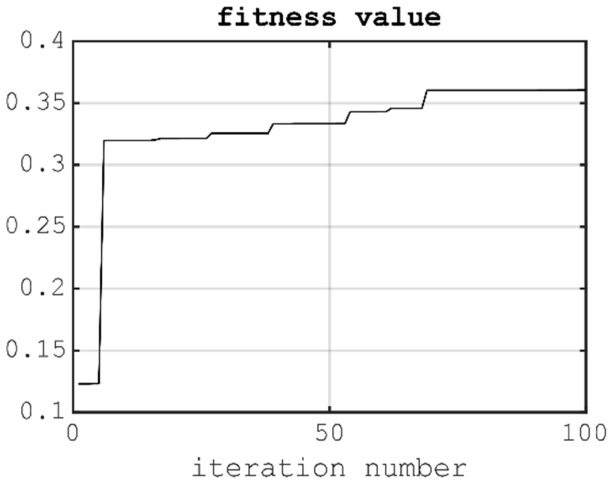

Figure 3. Values obtained by the fitness function of the genetic algorithm

Three criteria were taken into consideration when determining the fitness function of the genetic algorithm. These are the total errors in the step response of the system, settling time, and overshoot amount. The values these parameters assumed during optimization are shown in Figure 4.

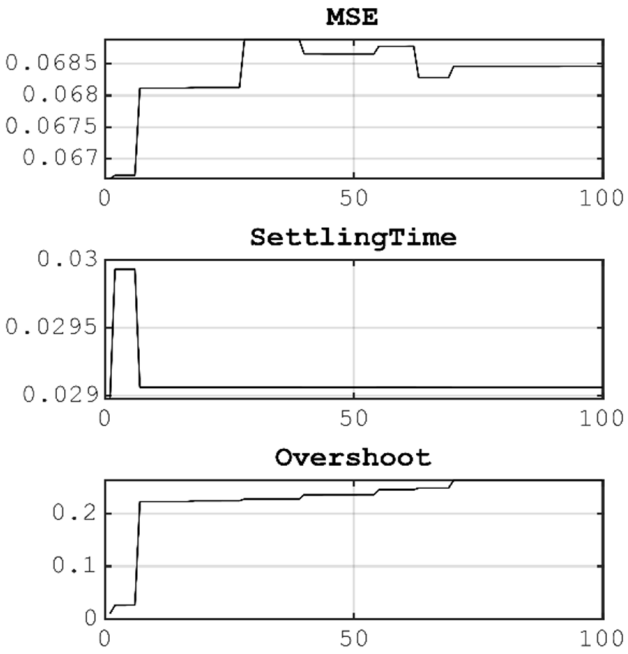

Figure 4. Variation of parameters constituting the fitness function

The PID coefficients obtained using the Ziegler-Nichols method were found to be kp=20, ki=.5, and kd=3. The response of the DC motor as a result of applying both the coefficients obtained with the genetic algorithm and those obtained with the Ziegler-Nichols method to the PID controller against a step reference signal with a set point=30 is presented comparatively in Figure 5. As can be seen from Figure 5, the genetic algorithm has found coefficients that yield better results.

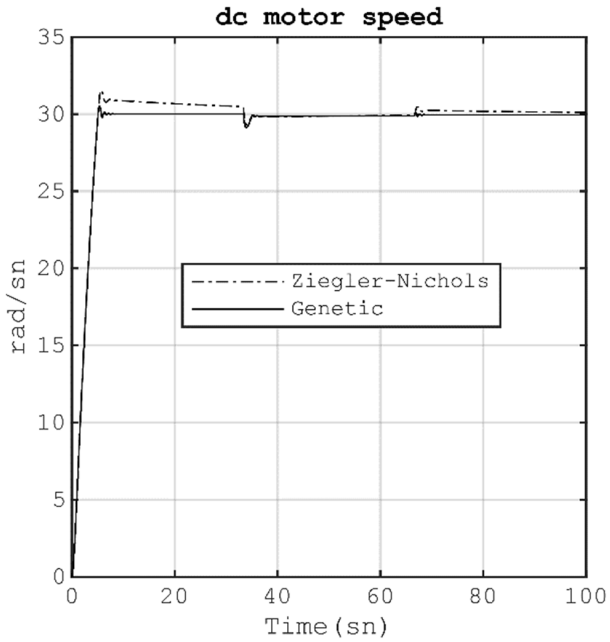

Figure 5. Step response of the DC motor

The results obtained through MATLAB's step info function are summarized in Table 4. According to Table 4, the genetic algorithm is better than Ziegler-Nichols. The overshoot is 1.992 with coefficients optimized by the genetic algorithm, while it is 3.8 with coefficients determined by the Ziegler-Nichols method. Therefore, the genetic algorithm has yielded better results than the Ziegler-Nichols method in terms of rise time, overshoot, peak, and peak time values.

Table 4. Comparison of DC motor speed results

Parameters	Ziegler-Nichols	Genetic
Rise time	3.800	3.799
Overshoot	4.724	1.922
Peak	31.536	30.533
Peak time	5.601	5.301

When the DC motor simulation is run with optimized coefficients, the changes in control voltage, motor current, angular velocity, derivative of motor current, angular acceleration, and load torque are shown in Figure 6. From this figure, it appears that the PID controller with coefficients optimized by the genetic algorithm successfully controls the angular velocity at the output despite changes in load torque.

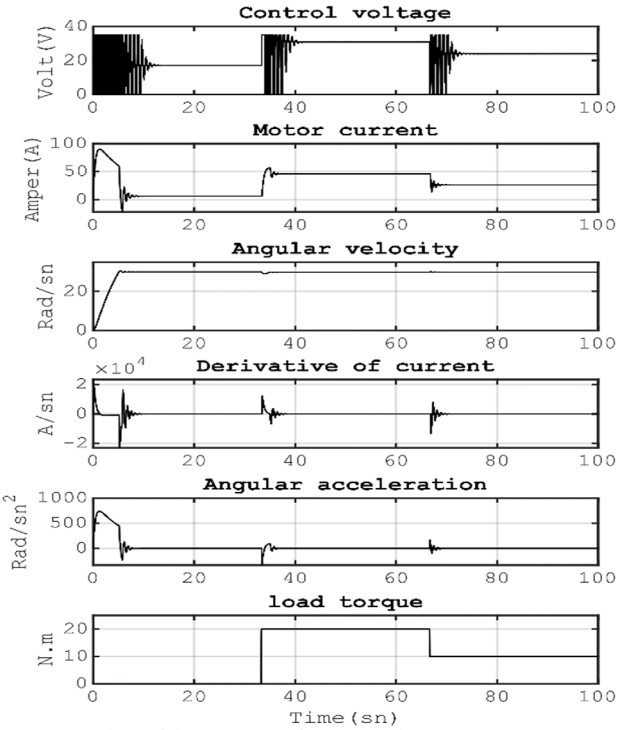

Figure 6. Values of the DC motor during simulation

CONCLUSION

As a result, this study has demonstrated the superior performance of PID controllers with coefficients optimized by genetic algorithms compared to those tuned by the traditional Ziegler-Nichols method for DC motor speed control. The research utilized a comprehensive mathematical model of a DC motor implemented in MATLAB/Script, incorporating realistic operating conditions such as time delay, voltage limitations, and variable load torques to simulate practical applications. The genetic algorithm optimization approach yielded PID coefficients of kp=95.984, ki=0.250, and kd=13.005, which demonstrated significant improvements over the Ziegler-Nichols method (kp=20, ki=0.5, kd=3) across all critical performance metrics. Notably, the GA-optimized controller reduced overshoot from 4.724% to 1.922%, while maintaining comparable rise time. The optimization

process incorporated multiple criteria in the fitness function, including total error in step response, settling time, and overshoot magnitude, ensuring a balanced control solution. The robustness of the GA-optimized controller was further evidenced by its excellent disturbance rejection capabilities. When subjected to sudden load torque variations of 20 N·m at 35 seconds and 10 N·m at 65 seconds, the controller maintained stable angular velocity, demonstrating its suitability for real-world applications where load conditions frequently change. This research contributes to the existing literature by providing a systematic comparison between conventional and evolutionary algorithm-based tuning methods for PID controllers in DC motor applications. The findings align with previous studies indicate that metaheuristic optimization techniques can significantly enhance control system performance beyond what traditional methods achieve.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Acharya, B. B., Dhakal, S., Bhattarai, A., & Bhattarai, N. (2021). PID speed control of DC motor using meta-heuristic algorithms. *Int J Power Electr Drive Syst*, *12*(2), 822-831. doi:10.11591/ijpeds.v12.i2.pp822-831

Ahangari Sisi, Z., Mirzaei, M., & Rafatnia, S. (2023). Time delay estimation and PID controller design using smith predictor for lever arm platform. *Iran J Mechan Eng Transact ISME*, *24*(2), 142-156. doi:10.30506/JMEE.2023.2011437.1323

Allaoua, B., Gasbaoui, B., & Mebarki, B. (n.d.). Leonardo Electronic Journal of Practices and Technologies Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy. Retrieved February 21, 2025, from http://lejpt.academicdirect.org

Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2021). A review of PID control, tuning methods and applications. *Int J Dynam Control*, *9*(2), 818-827. doi:10.1007/S40435-020-00665-4/FIGURES/1

Chauhan, S., Singh, B., Singh, M., & Li, Z. (2023). Review of PID control design and tuning methods. *J Physic Confer Series*, *2649*(1), 012009. doi:10.1088/1742-6596/2649/1/012009

de Figueiredo, R., Toso, B., & Schmith, J. (2023). Auto-Tuning PID Controller Based on Genetic Algorithm. In Disturbance Rejection Control. IntechOpen.

Gaing, Z. L. (2004a). A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transact Energy Convers*, *19*(2), 384-391. doi:10.1109/TEC.2003.821821

Gaing, Z. L. (2004b). A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transact Energy Convers*, *19*(2), 384-391. doi:10.1109/TEC.2003.821821

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms* (Vol. 1, pp. 69-93). Elsevier.

Idir, A., Kidouche, M., Bensafia, Y., Khettab, K., & Tadjer, S. A. (2018). Speed control of DC motor using PID and FOPID controllers based on differential evolution and PSO. *Int J Intell Eng Systems*, *11*, 4. doi:10.22266/ijies2018.0831.24

Jaen-Cuellar, A. Y., Romero-Troncoso, R. D. J., Morales-Velazquez, L., & Osornio-Rios, R. A. (2013). PID-controller tuning optimization with genetic algorithms in servo systems. *Int J Advan Robotic Systems*, *10*. doi:10.5772/56697/ASSET/IMAGES/LARGE/10.5772_56697-FIG12.JPEG

Jayachitra, A., & Vinodha, R. (2014). Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor. *Advanc Artific Intell*, *2014*, 1-8. doi:10.1155/2014/791230

Jigang, H., Jie, W., & Hui, F. (2018). An anti-windup self-tuning fuzzy PID controller for speed control of brushless DC motor. *Automat J Control, 58*(3), 321-335. doi:10.1080/00051144.2018.1423724

Kasilingam, G. (2014). Particle swarm optimization based PID power system stabilizer for a synchronous machine. *Int J Electr Comput Eng, 8*(1), 118-123.

Zahir, A. A., Alhady, S. S. N., Othman, W. A. F. W., & Ahmad, M. F. (2018). Genetic algorithm optimization of PID controller for brushed DC motor. *Lecture Note Mechan Eng*, *0*(9789811087875), 427-437. doi:10.1007/978-981-10-8788-2_38

Manuel, N. L., İnanç, N., & Lüy, M. (2023). Control and performance analyses of a DC motor using optimized PIDs and fuzzy logic controller. *Res Control Optimizat*, *13*, 100306. doi:10.1016/J.RICO.2023.100306

Ortatepe, Z. (2023). Genetic algorithm based PID tuning software design and implementation for a DC motor control system. *Gazi Univ J Sci Part A Eng Innovat*, *10*(3), 286-300. doi:10.54287/GUJSA.1342905

Patel, V. V. (2020). Ziegler-Nichols tuning method: understanding the PID controller. *Resonance*, *25*(10), 1385-1397. doi:10.1007/S12045-020-1058-Z

Pereira, D. S., & Pinto, J. O. P. (2005). Genetic algorithm based system identification and PID tuning for optimum adaptive control. *IEEE/ASME Int Confer Advanc Intell Mechatr, AIM*, *1*, 801-806. doi:10.1109/AIM.2005.1511081

Rothlauf, F. (2003). Representations for genetic and evolutionary algorithms. *J Operat Res Society, 54*(10), 1112.

Saravanan, G., Pazhanimuthu, C., & Naveen, P. (2025). Performance improvement of DC motor control system using PID controller with Kookaburra and Red Panda optimization algorithm. *Sci Rep, 15*(1), 20021. doi:10.1038/s41598-025-87607-2

Sharma, A., Sharma, V., & Rahi, O. P. (2022). PSO tuned PID controller for DC motor speed control. *Lecture Note Electr Eng*, *822*, 79-89. doi:10.1007/978-981-16-7664-2_7

Sivanandam, S. N., & Deepa, S. N. (2008). Genetic algorithm optimization problems. *Introduct Genet Algorit*, 165-209. doi:10.1007/978-3-540-73190-0_7

Tiwari, S., Bhatt, A., Unni, A. C., Singh, J. G., & Ongsakul, W. (2018). Control of DC motor using genetic algorithm based PID controller. *Proceed Confer Industr Commerc Use Energy, ICUE*, 1-16. doi:10.23919/ICUE-GESD.2018.8635662

Yadav, A., & Gupta, M. K. (2024). Design a controller based on smith predictor by direct synthesis method for speed control DC motor. *Int J Elect Eng Comp Sci*, *6*, 86-91. doi:10.37394/232027.2024.6.9

Zhang, J., Zhuang, J., Du, H., & Wang, S. (2009). Self-organizing genetic algorithm based tuning of PID controllers. *Informat Sci, 179*(7), 1007-1018. doi:10.1016/J.INS.2008.11.038

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0029

Performance comparison of compression algorithms for log data in compliance with Turkish Law No. 5651

Melih Karahan, Erdal Erdal, Atilla Ergüzen
Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kırıkkale University, Kırıkkale, Turkiye

Cite this article as: Karahan, M., Erdal E., & Ergüzen, A. (2025). Performance comparison of compression algorithms for log data in compliance with Turkish Law No. 5651. *J Comp Electr Electron Eng Sci, 3*(2), 54-60.

Received: 01.08.2025 • **Accepted**: 09.09.2025 • **Published**: 03.10.2025

ABSTRACT

Aims: The aim of this study is to evaluate the performance of various lossless compression algorithms for storing log data in compliance with Turkish Law No. 5651.

Methods: In the study, compression and decompression processes were performed on 100 MB and 10 GB of FortiGate log data that was generated synthetically from sample data using the Zstandard, Brotli, GZIP, LZ4, and Snappy algorithms. The algorithms were evaluated based on the amount of data compressed, compression ratio, time, speed, decompression time, and speed.

Results: The experimental studies reveal that the Zstd algorithm provides a balanced performance between compression ratio and speed. While the Brotli algorithm provides the highest compression ratio, it has the longest compression time and the slowest compression speed. The LZ4 and Snappy algorithms, on the other hand, have provided the best results in terms of compression speed and time, but their compression rates have lagged behind those of other algorithms.

Conclusion: The study results show that compression algorithms reveal significant differences in terms of performance and efficiency in large-scale log systems. The selection of the appropriate algorithm should be determined based on criteria such as the system's speed, storage requirements, and processing time. The findings obtained reveal that compression strategies play a critical role in fulfilling legal log retention obligations.

Keywords: Turkish Law No. 5651, big data, data storage, lossless compression, firewall logs

INTRODUCTION

Web 2.0 technology has significantly changed the way businesses, people, and communities interact through web-based environments and digital technology. One of the most important features of Web 2.0 technology is that it enables the widespread distribution of user-generated material. Additionally, the features associated with this technology have introduced various complexities. These features have led to new discussions about issues such as the legal management of user-generated materials, the protection of data privacy, and the oversight of produced content (Wang et al., 2024).

Data protection and privacy regulations (DPPR) have become widespread and continue to be developed worldwide. However, when the concept of the Internet and privacy is discussed, it becomes difficult to evaluate this issue as a whole and to manage it with a single law. Therefore, the laws on this subject have been classified under five main headings by the United Nations Conference on Trade and Development (UNCTAD). 195 countries around the world; 90% of countries have e-transactions laws, 79% of countries have data protection and privacy laws, 90% of countries have cybercrime laws, 60% of countries have consumer protection laws, and 50% of countries have indirect taxation laws (Unctad, 2025a). The laws and their respective status worldwide are shown in Table 1.

Table 1. Legal adoption status of cyberlaw areas worldwide

Law area	Legislation (%)	Draft legislation (%)	No legislation (%)	No data (%)
Electronic transactions	90	2	7	1
Data protection and privacy	79	3	17	1
Cybercrime	90	3	7	1
Consumer protection	60	2	29	9
Indirect taxation	50	0	50	0

Data sourced from Global Cyberlaw Tracker by UNCTAD (2025)

Turkiye's situation; when all these processes are examined for Turkiye, legislation exists for all the mentioned areas (Unctad, 2025b). The laws enacted for each aspect of the law in Turkiye and their details are presented in Table 2.

Although the five legal regulations presented in Table 2 cover the five specific topics identified by UNCTAD, they are insufficient for the collection and recording of Internet data. Therefore, a new law titled "Regulation on the procedures and principles regarding the regulation of publications made

Corresponding Author: Melih Karahan, karahanmelih@yahoo.com

Table 2. Laws adopted for each specific cyberlaw area in Turkiye

Law area	Name of law	Law no	Date
E-transactions	Electronic signature law	5070	23.01.2004
Data protection and privacy	Personal data protection law	6698	24.03.2016
Consumer protection	Law on consumer protection	6502	07.11.2013
Cybercrime	Turkish penal code Law	5377	29.06.2005
Indirect taxation	Communique No. 17 on VAT	3065	31.01.2018

Data sourced from Cyberlaw Tracker: the case of Turkiye by UNCTAD (2025)

on the internet environment" (Law No. 5651) was enacted (Republic of Turkiye, 2007). The aim and scope of this law are to define the duties and liabilities of content, hosting, access, and mass use providers and to set the rules and procedures for addressing crimes committed on the Internet via these providers.

Under the specified law, content may be removed from websites, and, if necessary, internet traffic data may be requested from the relevant authorities. Internet traffic data includes information such as the parties involved, time, duration, type of service used, amount of data transferred, and connection points related to Internet access. Article 6 further stipulates that access providers are required to store the specified traffic data for a minimum of six months and a maximum of two years and to ensure the accuracy, integrity, and confidentiality of this data. However, as these logs accumulate rapidly, especially in big data environments, they pose significant challenges in terms of storage, retrieval, and cost efficiency.

Data compression plays a crucial role in managing large-scale log repositories by significantly reducing disk usage while retaining valuable information for subsequent analysis and security audits. Selecting the right compression strategy directly impacts both storage efficiency and the operational costs associated with long-term log data retention (Li et al., 2024). Various compression algorithms, such as GZIP, LZ4, Snappy, Zstandard (Zstd), and Brotli, have been extensively studied for their effectiveness in general-purpose data compression. These algorithms are also widely adopted in key–value store systems to handle large volumes of unstructured data efficiently (Jaranilla and Choi, 2023). However, the performance of these algorithms on log data generated under specific legal constraints, such as those imposed by Turkish Law No. 5651, has not been sufficiently explored.

Literature Review

Big data: The concept of big data describes large data sets that are difficult or nearly impossible to manage using traditional analysis and data storage techniques (Raban and Gordon, 2020). The Big Data concept is frequently described using the four Vs: velocity, variety, volume, and veracity (Butts-Wilmsmeyer et al., 2020). However, the big data concept has recently been characterized by 7V (Moharm, 2019), as detailed in Table 3.

Big data storage: Although database systems continue to evolve in the big data era, the growing data volume generates challenges for many industries. When data sizes reach petabyte levels, databases experience performance and efficiency problems, despite optimization techniques (Chen et al., 2022).

Table 3. Big data 7Vs and details

Character	Basic description
Value	The valuable information when the data is properly analyzed
Volume	The actual amount of data collected and stored
Variety	The data is in an unstructured form or a mixture of structured and unstructured data
Velocity	New data values are generated at frequent intervals, usually in a constant flow
Veracity	The obtained data be healthy, that is, the uncertainty in its accuracy
Visualization	The visualization and readability of this obtained data
Variability	The variability of this data obtained

Data sourced from Moharm, K. (2019)

Traditional storage methods are inadequate in terms of performance and scalability. Therefore, various data storage technologies and architectures have been reshaped according to the needs of big data (Theodorakopoulos et al., 2024).

Across these technologies, Hadoop represents an open-source version of the cloud computing infrastructure developed by Google. Hadoop Distributed File System (HDFS) is a distributed file system that runs on standard hardware. The number of clusters can be increased as needed to meet the application requirements. Data are stored in block-structured small files. Its primary advantages include high scalability, reliability, and fault tolerance (Jing et al., 2018).

Additionally, while NoSQL does not have a precise definition, it refers to a database paradigm that offers non-relational solutions and avoids SQL (de Oliveira et al., 2021). There are four types of NoSQL approaches: Document-based, which stores each piece of data in an independent document. This structure is especially used for accessing, storing, and processing semi-structured data (e.g., MongoDB). Key-value-based, where each piece of data is kept in the form of a unique key and key-value pair, and these pairs are organized through a hash table. These values can be text, numbers, HTML, XML, JSON, videos, and images (e.g., Redis). Column-based, each data item is recorded in cell structures positioned within columns. These columns are classified into groups called column families to maintain a logical structure (e.g., Apache Cassandra). In graph-based data representation and storage operations, relationships and nodes that show the interaction between data are used (Ahmed et al., 2018).

Furthermore, cloud computing is a model that enables the on-demand and convenient distribution of computing resources such as storage, servers, applications, networks, and services over the Internet, with easy management and minimal involvement from the service provider (Rashid and Chaturvedi, 2019). In the public cloud, enterprises externally store their data by using services provided by cloud storage providers (e.g., Alibaba Cloud and Amazon Web Services) without establishing their infrastructure. It is preferred because of its scalability, low cost, and flexibility. The personal cloud is a type of public cloud, but it provides services for individual users. In the private cloud, enterprises establish and manage the infrastructure within their environment, which is managed by professional IT teams. Data are completely under the organization's control and are highly secure. However, due to the high cost, this is more convenient for storing sensitive data. The hybrid cloud is a mix of private and public cloud

storage. Sensitive data is stored in the private cloud, while other data is stored in the public cloud. It offers a flexible and balanced solution for enterprises. The community cloud is designed to address the shared requirements of several enterprises within the same sector (e.g., finance). The costs of infrastructure and management duties are distributed among community members (Yang et al., 2020).

Data compression: Compression techniques are categorized into two types; lossless and lossy methods. Lossy compression refers to a technique in which some portion of the original data is lost during the process. In some cases, such as real-time video streaming, a limited amount of data loss, such as reduced graphics or color depth, can be tolerated. This type of compression is applied to multimedia formats including JPEG, MP3, MPEG, and various video formats. Lossless compression is a method that decreases the size of data without losing quality or causing damage. When the data compressed with this method is restored, the original data decompresses precisely. It is widely used in file types such as spreadsheets and text documents, where data integrity is critical. While lossless compression is very successful in terms of quality, lossy compression can reach better compression ratios. Lossless compression techniques can be commonly categorized into two types; entropy-based encoding and dictionary-based encoding (Gupta and Nigam, 2021).

The Huffman coding algorithm is one of the entropy-based compression techniques and is named after D.A. Huffman, who developed the algorithm in the 1950s. The Huffman compression algorithm forms the basis of the most common algorithms and is implemented behind many programs, such as JPEG and GZIP. When the symbols that need to be coded and the possibility of their occurrence are calculated, this algorithm minimizes the number of bits required, thus providing optimal coding. Shorter codes are assigned to appear more frequently in characters, while longer codes are assigned to appear less frequently in characters.

The algorithm is created by the combination of the lowest characters, and this process only remains until the probability of only two combined characters is formed. Consequently, the Huffman tree is generated, and the code tree yields Huffman (Baidoo, 2023).

Arithmetic coding is a near-optimal coding technique that falls into the entropy coding category. Arithmetic coding involves assigning a decimal number in the range [0, 1] to a text. Both encoding and decoding processes are based on recursive range division. The encoder takes a character string and an expected probability distribution as input and produces a compressed structure. The decoder takes the compressed structure and expected probability distribution as input and produces the original string. The expected probability distribution determines the entropy and therefore affects the compression efficiency (Liu et al., 2019).

The run length encoding algorithm, an entropy-based coding algorithm, is the most basic of data compression algorithms. In this approach, consecutive character sequences are called "runs," while all other sequences are classified as "non-runs." This algorithm handles repeated characters. For example, the string "CDCDDDDE" acts as a source for compression; the first three letters are classified as non-runs of length three, while the next four letters are defined as a run of length four due to the repetition of the D character. The main purpose of this algorithm is to detect runs in the source file and document the character and length of each run. This algorithm compresses the runs in the original source file but does not compress the non-runs (Kodituwakku and Amarasinghe, 2010).

The Lempel-Ziv algorithm, which is a dictionary-based coding algorithm, was developed by the first members of the replacement compression, Abraham Lempel and Jakob Ziv, in 1977. LZW works with practically all kinds of data as a compression technique. LZW compression creates a structure of frequently occurring strings within the compressed data, replacing the real data with pointers to this structure. The same dictionary is created during compression and decompression, ensuring that the original data is accurately restored. LZW compression does not perform text analysis. Instead, it simply adds each new character sequence encountered to a set of character structures and replaces the character with a single code. Compression starts with a "dictionary" containing all single characters indexed from 0 to 255. Afterwards, it initiates the process of expanding the dictionary during data transmission. Duplicate strings will be compressed into a single byte; hence, compression is completed (Btoush and Dawahdeh, 2018).

The LZ77 algorithm uses the probability of repeating phrases and words in a text to compress data. Repeated data is represented by a number indicating the match length and a pointer pointing to a previous location. This method is a very simple, source-neutral, and adaptable compression method. In LZ77, the dictionary is obtained from a specific part of the previously encoded data sequence. The encoder uses a sliding window system structure to scan the data sequence. This window consists of two parts: the look-ahead buffer-the new data to be encoded-and the search buffer-the previously encoded data. This method searches for the longest match with the beginning of the data, and if a match is found, it indicates this match as <length, offset, char>. If no match is found, it outputs <0.0, char>. Length and offset values are defined using specific bit lengths. Usually, the match length is 8 bits, and the offset length is between 12 and 16 bits. The compression ratio is directly affected by these limits. Since multiple symbols are represented by a single reference, compression processes are rapid. However, searching for the longest match takes a long time. The decompression process is relatively fast by directly replacing each reference (Senthil and Robert, 2011).

METHODS

Dataset Description

The datasets used in this study consist of web traffic logs from the FortiGate firewall produced in compliance with Law No. 5651. This log data explains port scanning behaviors and is taken from the port scanning example presented in the source (Security, 2023). Although synthetic data was used, it was generated entirely from real FortiGate log structures, ensuring that all records preserved the exact format and field patterns of operational data. Log records include fields such as timestamp, source and destination IP addresses, requested URL, response size, HTTP method, and status code.

The datasets, the 10 GB dataset and the 100 MB dataset, have been created in plain text (TXT) format, with each line representing an HTTP request. These data were synthetically generated using the Java programming language and do not contain any personal information. In the synthetic data

generation process, fields such as srcip, dstip, srcport, dstport, duration, sentbyte, rcvdbyte, service, sessionid, and time were created randomly but within meaningful ranges, referencing the FortiGate device log structure. In order to replicate real network traffic, timestamps have been generated in a sequential and consistent manner; port and IP information have been configured according to the dynamics of internal and external networks. Some fixed identifier fields (devid, vd, policytype, etc.) have been used without modification. All compression algorithms were run on the same datasets under the same conditions.

Compression Algorithms

Zstandard (Zstd): Zstd is a lossless data compression algorithm developed by Meta. It can operate at fast execution speeds and high compression ratios (Facebook, 2025a).

In this study, the open-source library zstd-jni was used for compression and decompression via Java Maven. The code structure of Zstd includes both Huffman and FSE (Finite State Entropy)-based entropy structures. There are compression levels from 1 to 22. The default compression level of 3 was used in this study. Higher levels provide a higher compression ratio, while lower levels allow for faster results (Facebook, 2025a; Facebook, 2025b). Additionally, Zstd has dictionary-based compression support that generates efficient results in compressing repetitive data structures or small files (Facebook, 2025b). Zstd's compression format supports streams, allowing it to compress data without storing it in memory and to process it in parts. This functionality makes Zstd useful for applications requiring real-time data streaming or low-memory systems (Facebook, 2025a).

Brotli: Brotli is a modern algorithm developed by Google that provides lossless data compression. It uses dictionary-based (LZ77), entropy-based (Huffman) approaches, and second-order contextual modeling together. Brotli divides the text into small blocks, and these divided blocks are compressed separately (Prayogo et al., 2024). Brotli's encoding structure consists of dynamic Huffman coding, a static dictionary, context modeling, and a sliding window algorithm. Huffman tables are dynamically generated based on the compressed data. Brotli can also provide highly efficient results, especially for small-sized files and frequently repeated data structures, thanks to its support for static dictionaries (IETF, 2016).

The Brotli algorithm offers compression levels ranging from 0 to 11. Low levels (0-5) provide rapid compression; high levels (6-11) offer higher compression rates but increase processing (Google, 2025a). In this study, the open-source library brotli4j was used for compression and decompression via Java Maven. In the compression processes, the default compression level for Brotli, which is 5, was used.

GZIP: ZIP is an open-source algorithm used for file compression. This approach, created by Mark Adler and Jean-loup Gailly, is based on the LZ77 algorithm, which results in lossless data compression. Adding Huffman coding improves the results. GZIP divides the data into blocks, and each block is encoded separately. This format provides effective results, especially in text files. The GZIP compressed file contains not only the compressed data but also the filename and the date and time information (GNU Project, 2025; Syed and Soomro, 2018).

In this study, java.util.zip, which is part of the Java SDK, was used for compression and decompression operations in the Java environment without requiring external dependencies. The GZIP algorithm offers compression levels ranging from 1 to 9. Low levels (1-3) offer rapid compression but with a low compression ratio; high levels (4-9) provide slow compression but with a high compression ratio (Syed and Soomro, 2018). The default compression level has been used (this level is 6).

LZ4: LZ4 is a dictionary-based data compression technique that operates on the principle of identifying repeating trends in data sets and encoding them with the references to previous locations. This technique provides remarkable compression efficiency, especially in large datasets. A key characteristic of the algorithm is that it offers the user flexibility between compression ratio and speed. Depending on the application requirements, the user can choose to increase the compression speed at the expense of the compression ratio, or vice versa. As a member of the LZ4 family, LZ4HC (High Compression) offers better compression ratios by using higher processing power. This variant is particularly suitable for scenarios where compression efficiency is more important than speed (Jaranilla and Choi, 2023).

In this study, the open-source Java library lz4-java was used for the LZ4 algorithm. No adjustments were made regarding compression; it was used as is. Compression and decompression operations have been carried out via Maven.

Snappy: Snappy is a fast and lightweight data compression library developed by Google and offered as open source on GitHub. This algorithm is designed with a focus on speed rather than a high compression ratio (Google, 2025b). While working with dictionary-based techniques based on LZ77, it also includes some Huffman coding elements. Each compressed block may optionally contain a checksum for verification purposes. One of the most important advantages of Snappy is its outstanding performance in both compression and decompression processes (Jaranilla and Choi, 2023).

In this study, the Snappy algorithm was implemented using the open-source official library snappy-java integrated with the Java language. No special configuration was made for the compression processes; the default settings were used. Compression and decompression operations were performed using Maven.

All compression algorithms in this study were used with their default settings (Zstd at level 3, Brotli at level 5, GZIP at level 6, and no manual tuning for LZ4/Snappy). This choice ensures a fair comparison and avoids potential imbalances that could arise from using different tuning levels. Although higher or lower compression levels could reveal additional trade-offs between speed and compression ratio, the purpose of this study was to evaluate baseline performance under standard conditions.

RESULTS

Experimental Settings

All experiments were performed on a machine operating Windows 10 (22H2) with an intel (R) core (TM) i7-7500U CPU [2.70 gigahertz (GHz)], 16 gigabytes (GB) of RAM, and 256 GB solid-state drive (SSD) storage. The algorithms were executed using IntelliJ IDEA 2024 (community edition 2.1) with Java version 21.0.7. For compilation and configuration,

the maven-compiler-plugin version 3.11.0 was used, along with the central repository at repo1.maven.org. The compression libraries utilized in this study are zstd-jni version 1.5.7-3 for Zstandard, brotli4j version 1.18.0 for Brotli, java.util.zip.* from JDK 21 for GZIP, lz4-java version 1.8.0 for LZ4, and snappy-java version 1.1.10.1 for Snappy.

Dataset Specifications

The format of each log record is plain text, containing space-separated key-value pairs. For the experimental studies, synthetic datasets containing 172,270 log records (100 megabytes) and 17,639,965 log records (10 gigabyte) were utilized.

The content of the log records created by modification includes the following: The main time information includes codedate (YYYY-MM-DD) and time (HH:MM:SS), representing the log's creation timestamp, and it has been created to simulate a time distribution of approximately 27.7 hours (up to 100,000 seconds before creation). Device-specific identifiers, such as devname (e.g., FORTIGATE-02132, containing a random 5-digit number) and devid (e.g., FGT5510460331346, containing a random 13-digit number), uniquely identify the logging device. Each record also contains a unique logid.

Network addresses and ports are detailed with srcip and dstip (randomly generated from the internal ranges 192.168.x.x and the private range 10.0.x.x) and srcport and dstport (randomly generated in the range 1024-61023). Interface information is provided by randomly selecting srcintf and dstintf from internal, dmz, wan1, and vpn. Geographic contexts are captured with dstcountry and srccountry randomly selected from reserve, Turkiye, Germany, the USA, and the United Kingdom. The operating system version (OSVersion) is randomly selected from Microsoft Windows 10, Windows Server 2016, Linux, and Ubuntu 20.04. Finally, to complete the network context, MAC addresses (mastersrcmac, srcmac, dstmac) are generated randomly.

Experimental Results

This section presents the results of a comparison of five compression algorithms; Zstandard (Zstd), Brotli, GZIP, LZ4, and Snappy. The evaluation focuses on six performance metrics: compressed output size, compression ratio, compression time, decompression time, compression speed, and decompression speed. The experiments were conducted using synthetic FortiGate log files with sizes of 100 MB and 10 GB. Table 4, 5, and Figure display the results.

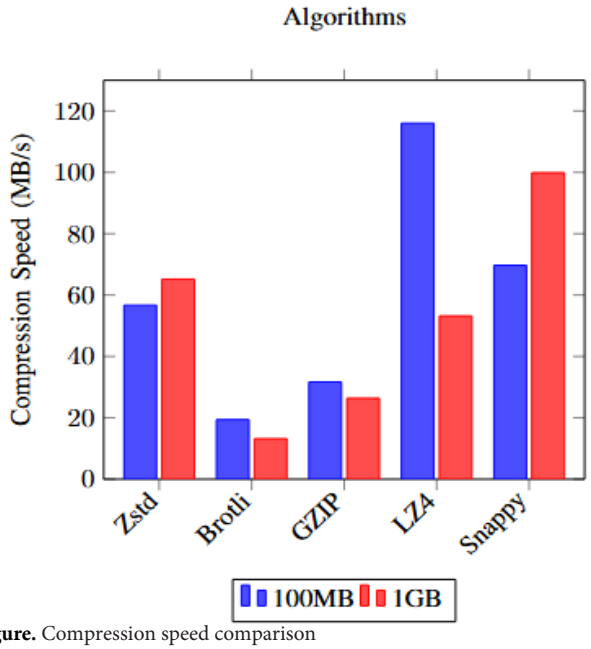

Figure. Compression speed comparison

Blue bars represent the 100 MB log dataset; red bars represent the 1 GB log dataset

DISCUSSION

In this study, the performances of different compression algorithms on 100MB and 10GB sized log data were compared. The results obtained have shown that the algorithms provide similar compression rates regardless of the data size, but their processing times and speeds vary according to the data size. Although including system performance metrics such as CPU and memory usage would have made the study stronger, the datasets used were archival data, and the primary focus of this study was on compression performance; therefore, CPU and memory usage metrics were considered out of scope.

Table 4. Compression performance results for the 100MB FortiGate log dataset

Algo	Comp size (MB)	Comp ratio (%)	Comp time (ms)	Decomp time (ms)	Comp speed (MB/s)	Decomp speed (MB/s)
Zstd	15.10	84.9	1765	541	56.7	184.8
Brotli	13.51	86.5	5142	454	19.4	220.3
GZIP	14.53	85.5	3154	1100	31.7	90.9
LZ4	25.68	74.3	862	373	116.0	268.1
Snappy	28.60	71.4	1435	475	69.7	210.5

Algo: Algorithm, Comp: Compression, Decomp: Decompression, MB: Megabyte, %: Percentage, ms: Milliseconds, MB/s: Megabyte per second

Table 5. Compression performance results for the 10GB FortiGate log dataset

Algo	Comp size (MB)	Comp ratio (%)	Comp time (ms)	Decomp time (ms)	Comp speed (MB/s)	Decomp speed (MB/s)
Zstd	1547.67	84.9	106463	53267	96.2	192.2
Brotli	1383.79	86.5	350703	49386	29.2	207.3
GZIP	1488.24	85.5	248491	79719	41.2	128.5
LZ4	2631.93	74.3	48950	57994	209.2	176.6
Snappy	2930.79	71.4	48476	58682	211.2	174.5

Algo: Algorithm, Comp: Compression, Decomp: Decompression, MB: Megabyte, %: Percentage, ms: Milliseconds, MB/s: Megabyte per second

The compression ratios obtained from Table 4 and 5 show significant differences in the data compression capabilities of the algorithms. In both datasets, the Brotli algorithm achieved the highest compression ratio (86.5%), followed by the GZIP (85.5%) and Zstd (84.9%) algorithms. These rates indicate that the compression algorithms provide similar levels of efficiency regardless of the data size. On the other hand, LZ4 and Snappy achieved lower compression rates (74.3% and 71.4%). However, the main advantage of these algorithms lies in their processing speeds.

Compression time and speed are important metrics in systems with real-time log processing or high data throughput. Table 4, 5, and Figure demonstrate the high compression speeds offered by the LZ4 and Snappy algorithms. In the 100 MB dataset, LZ4 achieved the fastest compression speed of 116 MB/s, followed by Snappy with 69.7 MB/s. Additionally, despite Zstd (56.7 MB/s), Brotli (19.4 MB/s), and GZIP (31.7 MB/s) having high compression ratios, their compression speeds remain slow.

In a 10 GB dataset size, there has been a decrease in the compression speeds of the algorithms, and the compression times have increased. Snappy achieved 211.2 MB/s as the fastest algorithm, while LZ4's compression speed fell to third place at 209.2 MB/s. Brotli, although it has a high compression ratio, is considered the slowest algorithm. It has 100 MB of data compressed in a time of approximately 5.1 seconds and around 350.7 seconds for 10 GB of data.

When evaluated in terms of decompression times, LZ4 achieved the fastest decompression with 268.1 MB/s (100 MB data) and 176.6 MB/s (10 GB data). Snappy, Brotli, and Zstd have also demonstrated high-speed decompression performance. However, GZIP has a low decompression speed in both datasets.

According to these results, LZ4 and Snappy are suitable for applications that require high speed despite low compression ratios. Zstd has been the most balanced algorithm in terms of speed and compression ratio. Brotli is well-suited for situations that require high-density compression and are not time-sensitive. GZIP has not been able to provide superiority in either speed or compression ratio.

Limitations

This study evaluated the performance of compression algorithms using raw log data.

Due to legal restrictions in Turkiye, the use of real production data was not possible. However, when deployed in a production environment, the same compression methods will be applied directly to real log streams. The experiments were conducted on datasets of 100 MB and 10 GB to ensure manageable yet non-trivial test sizes; however, extending the evaluation to larger datasets (e.g., 1 TB, 100 TB) would provide stronger evidence of scalability and improve the generalizability of the results. In future work, testing with real operational log data (once legally permissible) and integrating benchmark studies or industrial application reports will further validate the findings and strengthen the practical relevance of this research.

CONCLUSION

This research assessed five lossless compression algorithms (Zstd, Brotli, GZIP, LZ4, and Snappy) were evaluated for the purpose of compressing log data that must be retained under Turkish Law No. 5651. Synthetic log datasets of 100MB and 10GB were used to compare the compression ratio, compression time and speed, and decompression time and speed metrics of each algorithm.

The results indicate that Zstd provides an optimal balance between compression ratio and speed, making it a strong candidate for general-purpose log compression tasks. Brotli achieved the highest compression ratio for both datasets but incurred significant processing time, making it more suitable for archival datasets where time is not critical. Contemporary alternatives are gradually approaching GZIP, which displayed modest performance in both compression ratio and speed. LZ4 and Snappy offered the fastest compression and decompression speeds, which makes them appropriate for real-time or high-throughput logging systems, albeit at the cost of lower compression ratios.

As a result of these comparisons, it has been observed that the algorithm to be used for compressing logs should be selected according to the system's needs. If storage space is limited, algorithms with a high compression ratio should be used; however, if data needs to be processed and updated in real-time, speed-priority algorithms should be used. As the data size increases, the performance difference between algorithms becomes more noticeable. Therefore, general solutions do not provide the desired performance in the large datasets produced as a result of Law No. 5651. Because obligations of 5651 introduce design constraints that are not typically addressed in general compression studies: (i) strictly lossless and deterministic decoding across software versions; (ii) efficient, auditable retrieval of narrow time windows; and (iii) the ability to deliver legally requested periods as single-file archives without sacrificing verifiability. Evaluating compression in this legal–operational setting therefore requires re-examining algorithm trade-offs with compliance-aligned metrics rather than generic throughput alone. For this reason, the necessity of developing a compression algorithm for texts at higher rates has been identified and targeted as future work. In addition, Hadoop and AWS have been considered as part of the planned future work to provide an optimized solution specifically addressing this problem.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

Acknowledgments

The author acknowledges the assistance of the QuillBot tool, which was used for improving the clarity and fluency of the English in this manuscript.

REFERENCES

Abdulazeez, F. A., Ahmed, I. T., & Hammad, B. T. (2024). Examining the performance of various pretrained convolutional neural network models in malware detection. *Appl Sci, 14*(6), 2614. doi:10.3390/app14062614

Ahmed, M. R., Khatun, M. A., Ali, M. A., & Sundaraj, K. (2018). A literature review on NoSQL database for big data processing. *Int J Eng Technol, 7*(2), 902-906. doi:10.14419/ijet.v7i2.12113

Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An automated vision-based deep learning model for efficient detection of android malware attacks. *IEEE Access, 10*, 2700-2720. doi:10.1109/ACCESS.2022.3140341

Baidoo, P. K. (2023). Comparative analysis of the compression of text data using Huffman, arithmetic, run-length, and Lempel Ziv Welch coding algorithms. *J Adv Mathemat Comput Sci, 38*(9), 144-156. doi:10.9734/jamcs/2023/v38i91812

Btoush, M. H., & Dawahdeh, Z. E. (2018). A complexity analysis and entropy for different data compression algorithms on text files. *J Comput Communicat, 6*(1), 301. doi:10.4236/jcc.2018.61029

Butts-Wilmsmeyer, C. J., Rapp, S., & Guthrie, B. (2020). The technological advancements that enabled the age of big data in the environmental sciences: a history and future directions. *Curr Opin Environment Sci Health, 18*, 63-69. doi:10.1016/j.coesh.2020.07.006

Chen, Y., Zhang, F., Hong, Y., et al. (2022). Taming the big data monster: managing petabytes of data with multi-model databases. Proceedings of the 2022 IEEE 34th International Symposium on Computer Architecture and High Performance Computing (pp. 283-292). IEEE.

de Oliveira, V. F., Pessoa, M. A. O., Junqueira, F., & Miyagi, P. E. (2021). SQL and NoSQL databases in the context of Industry 4.0. *Machines, 10*(1), 20. doi:10.3390/machines10010020

Facebook. (2025a). Zstandard (Zstd) [Computer software]. GitHub. Retrieved June 20, 2025, from https://github.com/facebook/zstd

Facebook. (2025b). *Zstandard API manual*. Retrieved June 20, 2025, from https://facebook.github.io/zstd/doc/api_manual_latest.html#Chapter1

Google. (2025a). Brotli compression tool manual page (brotli.1). Retrieved June 20, 2025, from https://github.com/google/brotli/blob/master/docs/brotli.1

Google. (2025b). Snappy - a fast compressor/decompressor [computer software]. GitHub. Retrieved June 20, 2025, from https://github.com/google/snappy

GNU Project. (2025). Gzip [Software]. GNU Operating System. Retrieved June 20, 2025, from https://www.gnu.org/software/gzip/manual/gzip.html

Gupta, A., & Nigam, S. (2021). A review on different types of lossless data compression techniques. *Int J Sci Res Comput Sci, 7*(1), 50-56. doi:10.32628/cseit217113

IETF. (2016). Brotli compressed data format (RFC 7932). https://datatracker.ietf.org/doc/html/rfc7932

Jaranilla, C., & Choi, J. (2023). Requirements and trade-offs of compression techniques in key-value stores: a survey. *Electronics, 12*(20), 4280. doi:10.3390/electronics12204280

Jing, W., Tong, D., Chen, G., Zhao, C., & Zhu, L. (2018). An optimized method of HDFS for massive small files storage. *Comput Sci Informat Syst, 15*(3), 533-548. doi:10.2298/CSIS170926034J

Kodituwakku, S. R., & Amarasinghe, U. S. (2010). Comparison of lossless data compression algorithms for text data. *Indian J Comput Sci Eng, 1*(4), 416-425.

Republic of Turkiye. (2007). Law No. 5651: Regulation on the procedures and principles regarding the regulation of publications made on the internet environment. https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=5651&MevzuatTur=1&MevzuatTertip=5

Li, X., Zhang, H., Le, V. H., & Chen, P. (2024, February). Logshrink: Effective log compression by leveraging commonality and variability of log data. In Proceedings of the 46th IEEE/ACM International Conference on Software Engineering (pp. 1-12).

Liu, Q., Xu, Y., & Li, Z. (2019, February). DecMac: a deep context model for high efficiency arithmetic coding. In 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 438-443). IEEE.

Moharm, K. (2019). State of the art in big data applications in microgrid: A review. *Adv Eng Informat, 42*, 100945. doi:10.1016/j.aei.2019.100945

Prayogo, A. I., Nugraha, A., Kurniawan, J. C., Kusuma, E. J., & Wibowo, A. (2024). Enhancing least significant bit steganography image fidelity using Brotli compression. *Sinkron J Res Informat Technol, 8*(1), 285-295. doi:10.33395/sinkron.v9i1.13186

Raban, D. R., & Gordon, A. (2020). The evolution of data science and big data research: a bibliometric analysis. *Scientometrics, 122*(3), 1563-1581. doi:10.1007/s11192-019-03332-4

Rashid, A., & Chaturvedi, A. (2019). Cloud computing characteristics and services: a brief review. *Int J Comput Sci Eng, 7*(2), 421-426. doi:10.26438/ijcse/v7i2.421426

Security, R. (2023). Detecting attacks using Fortigate firewall logs with log samples [Blog post]. Medium. Retrieved June 20, 2025, from https://readsecurity.medium.com/detecting-attacks-using-fortigate-firewall-logs-with-log-samples-ad0e7ee57903

Senthil, S., & Robert, L. (2011). Text compression algorithms: a comparative study. *J Commun Technol, 2*(4), 444-451. doi:10.21917/ijct.2011.0062

Syed, Z., & Soomro, T. (2018). Compression algorithms: Brotli, Gzip and Zopfli perspective. *Indian J Sci Technol*, *11*(45), 1-4. doi:10.17485/ijst/2018/v11i45/117921

Theodorakopoulos, L., Theodoropoulou, A., & Stamatiou, Y. (2024). A state-of-the-art review in big data management engineering: real-life case studies, challenges, and future research directions. *Eng, 5*(3), 1266-1297. doi:10.3390/eng5030071

UNCTAD. (2025a). Summary of adoption of e-commerce legislation worldwide. United Nations Conference on Trade and Development. Retrieved June 20, 2025, from https://unctad.org/topic/ecommerce-and-digital-economy/ecommerce-law-reform/summary-adoption-e-commerce-legislation-worldwide

UNCTAD. (2025b). Cyberlaw tracker - Turkiye. United Nations Conference on Trade and Development. Retrieved June 20, 2025, from https://unctad.org/page/cyberlaw-tracker-country-detail?country=tr

Wang, S., Jiang, X., & Khaskheli, M. B. (2024). The role of technology in the digital economy's sustainable development of Hainan Free Trade Port and genetic testing: cloud computing and digital law. *Sustainability, 16*(14), 6025. doi:10.3390/su16146025

Yang, P., Xiong, N., & Ren, J. (2020). Data security and privacy protection for cloud storage: a survey. *IEEE Access, 8*, 131723-131740. doi:10.1109/ACCESS.2020.3009876

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0030

Personalized adaptive e-learning application based on learning styles

Ali Karay, Erdal Erdal, Atilla Ergüzen
Department of Computer Engineering, Faculty of Engineering and Natural Science, Kırıkkale University, Kırıkkale, Turkiye

Cite this article as: Karay, A., Erdal, E., & Ergüzen A. (2025). Personalized adaptive e-learning application based on learning styles. *J Comp Electr Electron Eng Sci, 3*(2), 61-67.

Received: 22.08.2025 • **Accepted**: 23.09.2025 • **Published**: 03.10.2025

ABSTRACT

This study presents the development of an adaptive e-learning system designed to deliver personalized content according to students' learning styles. The system analyzes individual learner profiles through a profiling module, structures learning resources via a content management module, and supports the process with instant feedback mechanisms. By dynamically adapting the flow of content, the system creates distinct learning journeys for students. The study presents the system's architecture, design-based research methodology, and its intended contributions to learner-centered education. It is anticipated that the developed framework will support future research in adaptive e-learning. Thus, the study is expected to provide useful insights into personalized learning processes at both the national and international level.

Keywords: Personalized e-learning, adaptive learning, learning styles, learner-centered education, educational technologies, design-based research, educational data mining s

INTRODUCTION

Digitalization in education has accelerated significantly in recent years, making student-centered teaching approaches increasingly important. Traditional e-learning systems are mostly based on standardized content delivery and fail to adequately consider students' individual learning differences. However, each student possesses a distinct learning style, motivation, and cognitive approach. This situation necessitates that e-learning platforms become more flexible, personalized, and adaptive (Ruman, 2022).

In this context, the system designed in this study aims to provide a personalized adaptive learning environment tailored to students' learning styles. By employing learner profiling techniques, the system collects data on individuals' learning preferences and processes this data to automatically adapt content delivery. For instance, visual learners are provided with diagrams, auditory learners with audio-based content, and kinesthetic learners with interactive applications.

The primary motivation of this research is to integrate the fact that the learning process differs for every student into the system infrastructure, thereby enabling more effective, lasting, and meaningful learning experiences. Furthermore, personalization based on learning styles enhances student engagement, strengthens learning outcomes, and increases motivation during the teaching process (Graf & Kinshuk, 2009).

THEORETICAL FRAMEWORK

Learning Styles

These styles refer to the individual differences in the ways people perceive, process, and recall information. In this study, the VARK model was adopted as the main reference for adaptation, due to its practicality and broad use in adaptive e-learning research. One of the most widely used models is the VARK model (visual, auditory, reading/writing, kinesthetic). According to this model:

- Visual learners prefer to learn through visual materials such as charts, graphs, and diagrams.
- Auditory learners learn by listening to lectures, participating in discussions, and following verbal instructions.
- Reading/writing learners benefit from text-based resources.
- Kinesthetic learners learn through hands-on activities, simulations, and experiential practices.

The model developed by Dunn and Dunn (1992) adopts a more comprehensive perspective by considering environmental, emotional, social, and psychological factors as well. Such models form the foundation of learner-centered systems.

Personalized Learning

The personalized learning approach aims to provide differentiated content pathways based on the individual characteristics of each student. In this approach, the system organizes the course flow according to the learner's current knowledge level, interests, and learning styles. Thus, a student may experience a distinct learning journey even within the same course.

Personalization strategies include learner profiling, content adaptation, dynamic assessment, and instant feedback.

Corresponding Author: Ali Karay, a.karay@yahoo.com

Particularly, artificial intelligence-based algorithms play a key role in optimizing the learner's pathway in real time (Romero & Ventura, 2020).

Adaptive E-Learning Systems

Adaptive systems are system structures that monitor learners' behaviors and adjust content, methods, or difficulty levels accordingly. The primary goal of such systems is to automatically select the most appropriate learning pathway for each individual learner.

In the literature, adaptive systems are defined by certain common features (Brusilovsky & Millán, 2007):

- User modeling (analyzing the learner's knowledge, skills, and interests)
- Content adaptation (selecting suitable materials for the learner)
- Task adaptation (assigning exercises and activities according to the learner's level)
- Navigation support (intelligent modules that guide learners)

The system developed in this research was designed on the basis of an adaptive architecture, with a specific focus on personalizing content delivery according to students' learning styles.

METHODS

An adaptive e-learning system was developed by redesigning all modules on an open-source learning management infrastructure with personalization and adaptivity at the core.

The system was primarily based on the VARK model, which is one of the most widely recognized frameworks in educational research. This model was selected because it provides clear categorization and supports multiple modalities of learning content. Thus, it allows the system to adapt resources to diverse learner preferences.

The VARK model was chosen because it is widely recognized in adaptive learning research, provides a clear categorization of learner preferences, and supports multiple modalities (visual, auditory, reading/writing, kinesthetic). This alignment ensures that the system can flexibly adapt content to diverse learning needs.

The development process followed the design-based research (DBR) methodology, which includes the following five systematic phases:

Analysis: Identification of the educational problem and justification of personalization based on learning styles.

Design: Planning of the learner profiling, content management, and feedback modules.

Development: Technical creation and integration of the modules.

Implementation: Pilot application of the system and collection of user feedback.

Evaluation: Continuous improvement of the system based on collected data.

This structured approach ensures the translation of theory into practice and enables iterative system refinement.

Components of the System

The developed system consists of three main modules:

- **Learner profiling module:** Analyzes the student's learning style, past performance, and preferences.
- **Content management module:** Structures materials provided by instructors and ensures that the most suitable resources are presented according to the learner's individual characteristics.
- **Assessment and feedback module:** Tracks learners' progress, provides instant feedback, and updates the content flow.

A continuous data flow is maintained among these modules, enabling the system to dynamically optimize the student's learning process.

SYSTEM ARCHITECTURE

In this study, an adaptive e-learning system was developed that takes into account students' learning styles. While the developed system is based on an open-source learning management infrastructure, all modules were redesigned to place personalized and adaptive learning processes at the core. Specifically, the system was developed on a Drupal-based open-source framework. Drupal was selected due to its modular architecture, extensive plugin support, customizable user roles, scalability, and integrated reporting features. These characteristics provided the flexibility required to implement adaptive and personalized functionalities.

The development process was carried out using the DBR methodology. This approach aims to translate.

This diagram shows the basic components of the developed adaptive e-learning system and the data flow between these components. The architectural structure consists of three main layers:

- **User layer:** Learners and instructors interact with the system, generating activity data.
- **Application layer:** Personalization engine, content management, task assignment, and reporting units analyze learner profiles and adapt the learning flow.
- **Data layer:** Stores learner profiles, performance data, and activity results to feed the application layer for ongoing adaptation.

Student interactions are first recorded in the user interface, then processed in the application layer to adapt content delivery and transfer it to reporting screens. Thus, the system personalizes the student's learning path and provides the teacher with detailed monitoring capabilities. The flow diagram of this structure is shown in Figure 1 below.

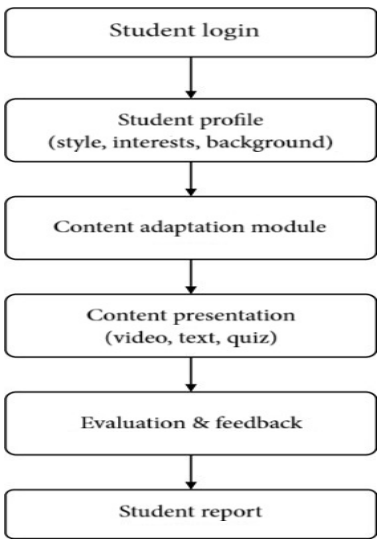

Figure 1. General system architecture (Karay & Erdal & Ergüzen, 2025)

The process of adaptive e-learning system is explained step by step, starting from the student's login to the system, through their interaction with the content, to the evaluation of learning outcomes:

- **Login and profile definition:** After the student logs into the system with their username/password, the system activates the student's profile (pre-test, learning style questionnaire, previous achievement data).
- **Learning style analysis:** The student's individual learning style (e.g., visual, auditory, kinesthetic) and current knowledge level are analyzed. This analysis serves as the starting point for content presentation.
- **Content selection and adaptation:** The most suitable content path is determined based on the student's profile data. For example, students with a visual learning style are presented with content heavy on graphics, diagrams, and videos, while those with an auditory style are prioritized with audio narrations.
- **Activity and task assignment:** The system automatically assigns activities and tasks appropriate to the student's level. The difficulty level is dynamically adjusted according to the student's performance.
- **Progress tracking and feedback:** The student's activity completion status, success rate, and learning time are tracked. The system provides instant feedback and guidance to the student.
- **Reporting and continuous adaptation:** The student's progress data is reflected in the teacher's dashboard. The student's progress is continuously analyzed, and new content and activities are adapted accordingly.

Figure 2 below illustrates the system 's personalization mechanism and concretizes the system's dynamic, student-centered operation.

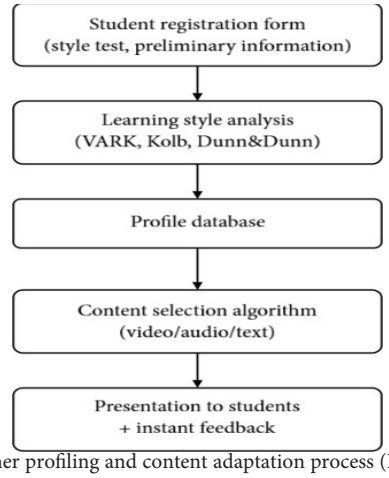

Figure 2. Learner profiling and content adaptation process (Karay & Erdal & Ergüzen, 2025)

SYSTEM FEATURES

Create New Account

A registration form is provided for new users, allowing them to easily create login credentials with basic information (first name, last name, and e-mail). A screenshot is shared in Figure 3.

Login Screen

Users logging into the system are authenticated with a username and password. For security purposes, a "Request New Password" option is also available on this screen. A screenshot is shared in Figure 4.

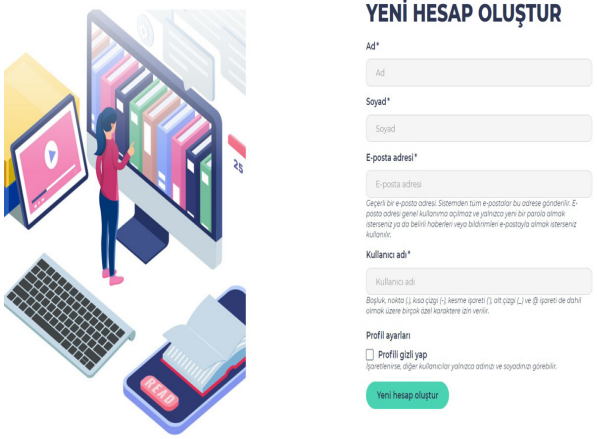

Figure 3. Create new account screenshot (Karay & Erdal & Ergüzen, 2025)

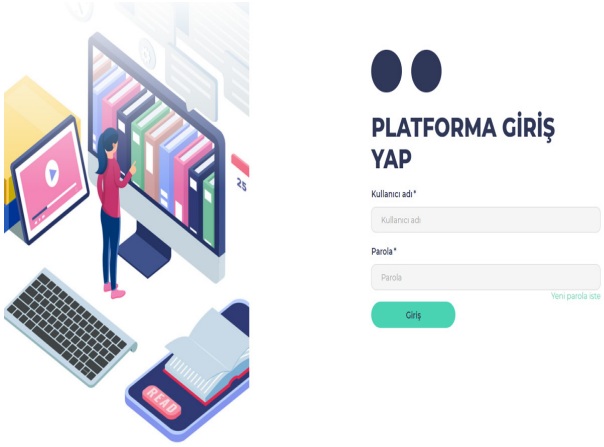

Figure 4. Login screen account screenshot (Karay & Erdal & Ergüzen, 2025)

Student Homepage

This screen displays students' current activities, calendar events, messages, and statistics within a single panel. In particular, social interaction features (comments, likes) enhance motivation in the learning process.

The interface of the system is designed to be simple, accessible, and mobile-friendly. After logging in, students are presented with a personalized main dashboard.

The system provides customizable dashboard blocks for each role (student/instructor/administrator); "my courses," "ongoing modules," "recent activities," "upcoming tasks," "achievement badges/certificates," "my reports." This approach offers users a task-oriented entry experience. A screenshot is shared in Figure 5.

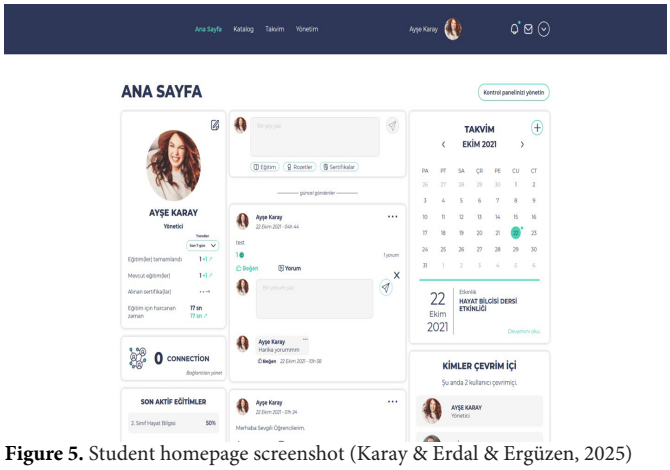

Figure 5. Student homepage screenshot (Karay & Erdal & Ergüzen, 2025)

Course Catalog Page

This screen allows students to view the courses they are enrolled in and monitor their progress. For each course, a visual, description, completion percentage, and a "continue learning" option are provided. In this way, students can quickly track how much of each course they have completed. A screenshot is shared in **Figure 6**.

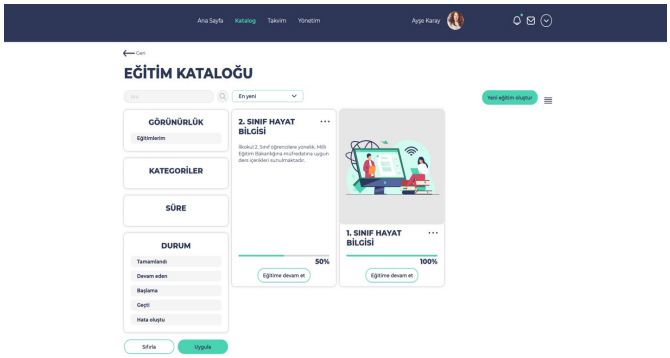

Figure 6. Course catalog page screenshot (Karay & Erdal & Ergüzen, 2025)

Course Detail Page-1

This screen allows students to view the courses they are enrolled in and monitor their progress. For each course, a visual, description, completion percentage, and a "continue learning" option are provided. In this way, students can quickly track how much of each course they have completed. A screenshot is shared in **Figure 7**.

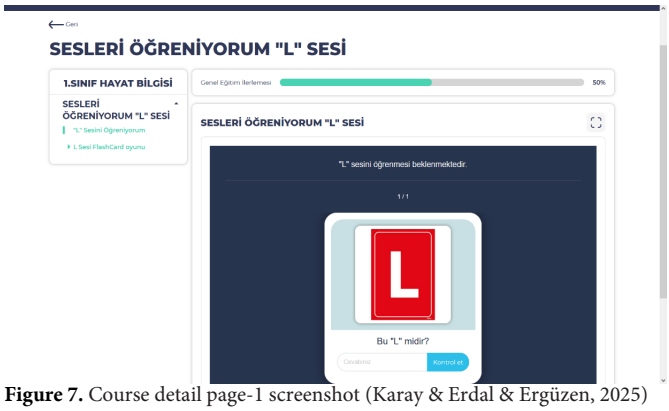

Figure 7. Course detail page-1 screenshot (Karay & Erdal & Ergüzen, 2025)

Course Detail Page-2

In the second course detail screen, a personalized feedback area and detailed information about completed activities are displayed. Students have the opportunity to evaluate their own performance and identify their areas of weakness. A screenshot is shared in **Figure 8**.

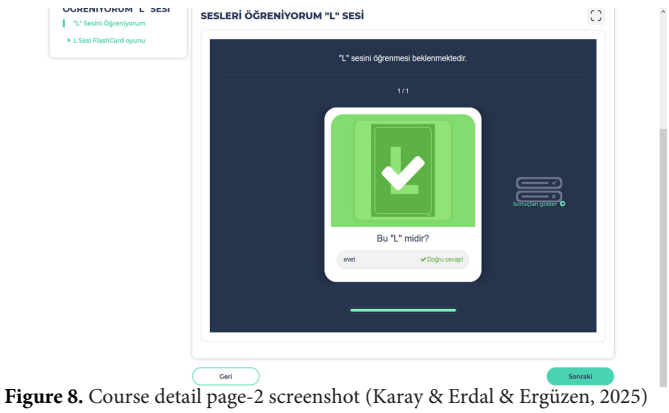

Figure 8. Course detail page-2 screenshot (Karay & Erdal & Ergüzen, 2025)

Course Detail Page-3

This screen contains the results section, which specifically summarizes the student's correct and incorrect answers. In this way, students can see not only their overall success rates but also identify in which activities they made mistakes. A screenshot is shared in **Figure 9**.

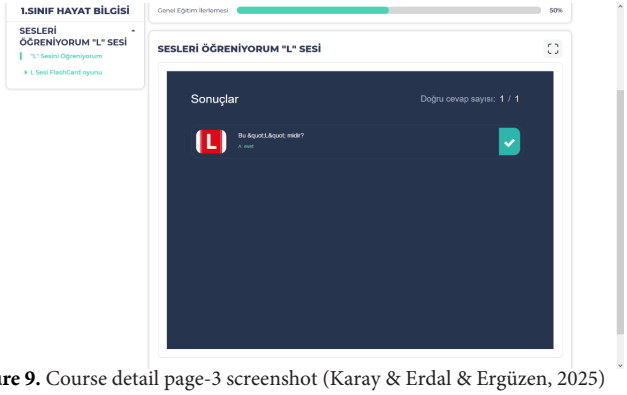

Figure 9. Course detail page-3 screenshot (Karay & Erdal & Ergüzen, 2025)

Learning Path Manager

Trainings are organized with a course → module → activity hierarchy through a drag-and-drop interface. With the tree view, modules within a course are visible, and new modules can be added or existing ones attached within a single screen. A screenshot is shared in **Figure 10**.

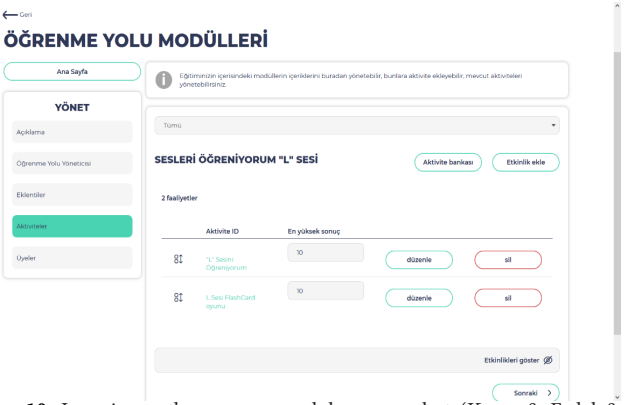

Figure 10. Learning path manager modules screenshot (Karay & Erdal & Ergüzen, 2025)

In designing the student module, the system was conceptually oriented toward elementary-level classes, focusing on basic literacy and numeracy activities. This level was considered appropriate because younger learners benefit strongly from multimodal content delivery (visual, auditory, and kinesthetic). However, this module has not yet been implemented in a live classroom setting; it remains a planned direction for future empirical testing.

Activities and Learning Modules

- **Course:** The pedagogical unit of the program; registration, access, and prerequisites are defined here.
- **Module:** Content packages (video, document, H5P, quiz, assignment, etc.) and sequencing rules.
- **Activity:** The atomic element with which the learner interacts (H5P interaction, SCORM package, quiz questions, file-upload assignments, etc.).

The system provides an enriched learning experience through H5P content types (interactive video, presentation, hotspot, timeline, etc.), and learner interaction data within the content (completion, score, attempts) can be monitored in real time.

Activity List View

Courses are supported with a variety of learning activities (quizzes, forums, SCORM content, video lessons, etc.), which students complete in order to progress.

The activity list presents all activities defined within a course in a tabular format. For each activity, the activity ID, highest score, edit, and delete options are displayed. This structure enables instructors to track learners' progress and update activity content when necessary. A screenshot is shared in **Figure 10**.

Adding Activities from the Activity Bank

This screen allows instructors to select from a variety of learning activities. For example, options such as file upload, long-answer questions, video, audio recording, slide presentation, or drag-and-drop are listed. This diversity supports the use of multiple sensory modalities in the learning process and addresses different learning styles. A screenshot is shared in **Figure 11**.

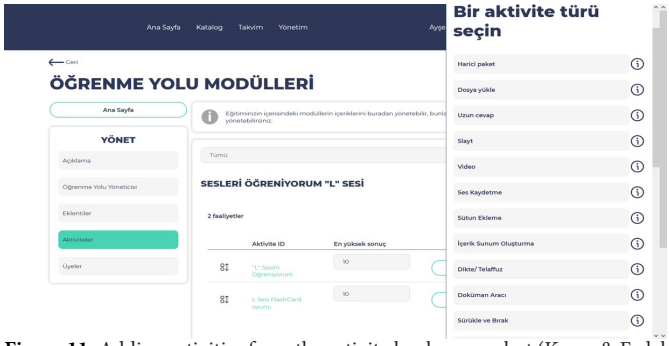

Figure 11. Adding activities from the activity bank screenshot (Karay & Erdal & Ergüzen, 2025)

Activity Type Selection Screen

This section, as a continuation of the previous screen, highlights assessment-focused activities such as fill-in-the-blank, flashcards, multiple-choice tests, timelines, and true/false questions. This module provides students with opportunities for reinforcement at both the cognitive and practical levels. A screenshot is shared in **Figure 12**.

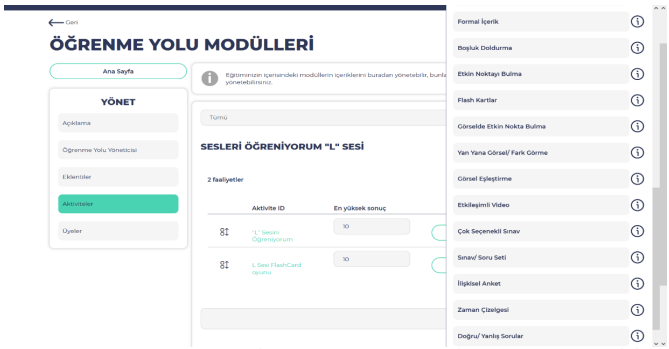

Figure 12. Activity type selection screen screenshot (Karay & Erdal & Ergüzen, 2025)

In-Class Messaging Screen

This screen includes a discussion area that enables real-time communication between instructors and students. Dedicated chat rooms for course groups support information sharing and social interaction among students and instructors. A screenshot is shared in **Figure 13**.

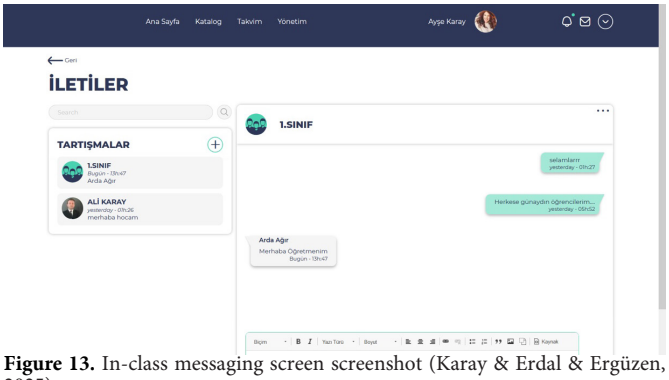

Figure 13. In-class messaging screen screenshot (Karay & Erdal & Ergüzen, 2025)

Student Profile Screen

The student profile displays identity information, enrollment date, last access time, earned badges, and enrolled courses. In addition, the trends section presents graphical data on the number of completed courses, time spent, and activity level within the last seven days. A screenshot is shared in **Figure 14**.

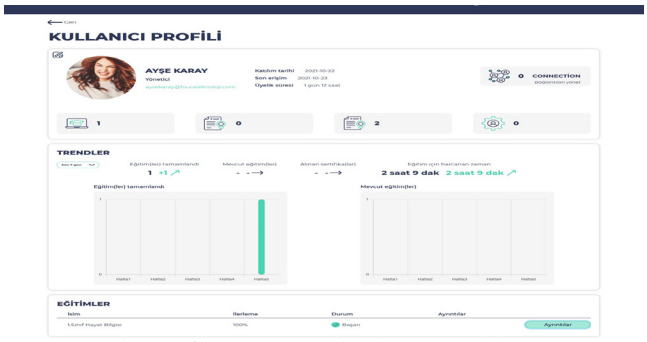

Figure 14. Student profile screen screenshot (Karay & Erdal & Ergüzen, 2025)

Student Achievement Screen

This screen lists the trainings completed by students, along with their success rates and statuses. For each course, details such as completion percentage, date, and results are presented comprehensively. In this way, students can clearly view their individual performance. A screenshot is shared in **Figure 15**.

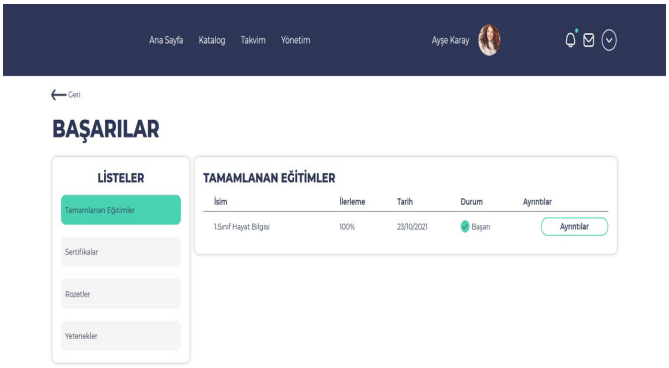

Figure 15. Student achievement screen screenshot (Karay & Erdal & Ergüzen, 2025)

Reporting and Statistic

Multi-layered reporting (global, course-level, user-level) allows monitoring of completion rates, average scores, number of attempts, activity duration, and time-series trends.

When certain activities are completed, skill achievements can be automatically linked; the logic of offering an "automatic skill module" based on target competencies is also supported. Certificates are generated once a course or learning path is completed.

User-Based Course Progress

A personalized reporting screen for each student details course-based progress, scoring, time spent, and earned badges. This enables students to monitor their individual development, while instructors can track student performance. A screenshot is shared in Figure 16.

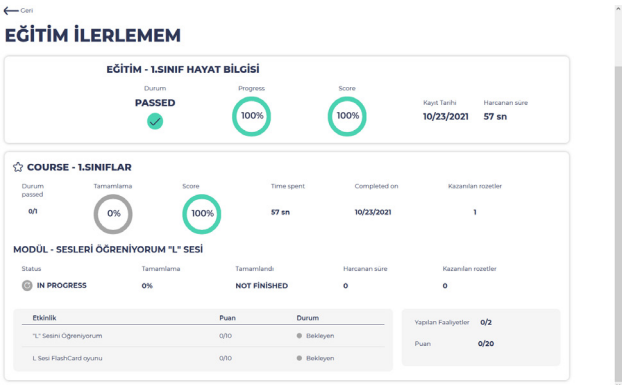

Figure 16. User-based course progress screen screenshot (Karay & Erdal & Ergüzen, 2025)

Training Statistics

This page presents summary statistics about the overall performance of the system in graphical form. Indicators such as learning progress percentage, completion rates, number of new users, and active users enable administrators to make data-driven decisions. A screenshot is shared in Figure 17.

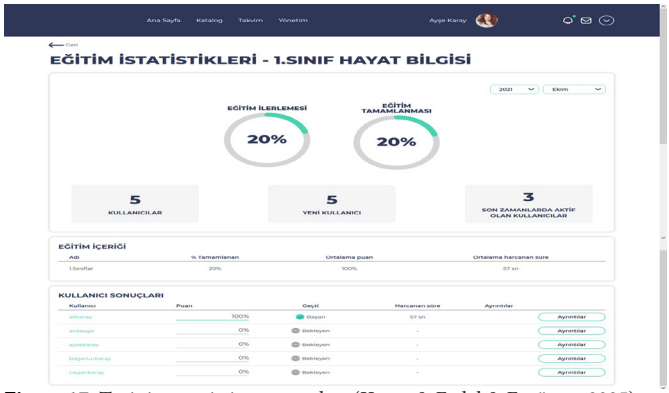

Figure 17. Training statistics screenshot (Karay & Erdal & Ergüzen, 2025)

School-Wide Statistics

These screens present institution-level data such as the number of users, course completion rates, distributions of active users, and overall success averages. Such statistics support decision-making at the administrative level and contribute to the improvement of educational policy. A screenshot is shared in Figure 18.

Event Creation and Calendar Management

The calendar module enables students and instructors to track course events on a date-based basis. Event start and end times, descriptions, and invited users can be displayed. This feature strengthens organization within the online learning process. A screenshot is shared in Figure 19.

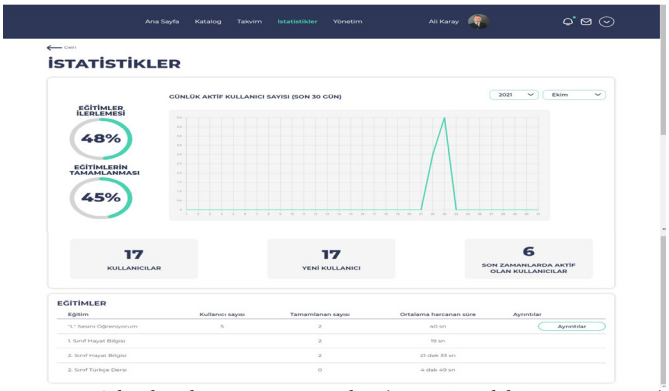

Figure 18. School-wide statistics screenshot (Karay & Erdal & Ergüzen, 2025)

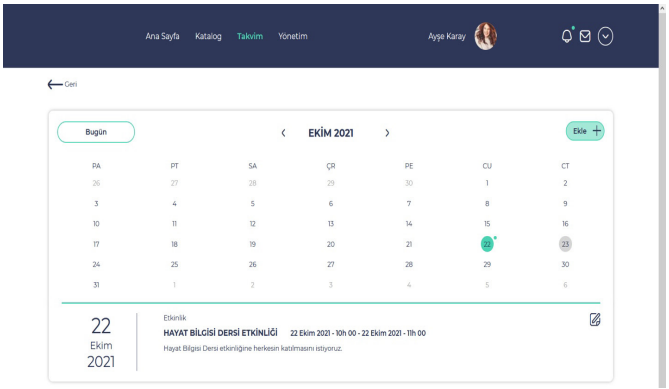

Figure 19. Event creation and calendar management screenshot (Karay & Erdal & Ergüzen, 2025)

Adding and Tracking Students in a Group/Class

Through this screen, education administrators can add new students to course groups, adjust the roles of existing members, and monitor participation dates. Group-based management helps instructors maintain control at the class level. A screenshot is shared in Figure 20.

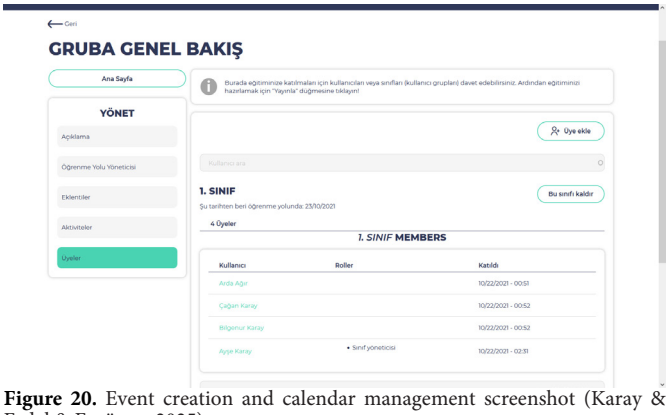

Figure 20. Event creation and calendar management screenshot (Karay & Erdal & Ergüzen, 2025)

DISCUSSION

Comparison with the Literature

Previous research (Romero & Ventura, 2020) emphasizes that adaptive systems improve student achievement, and our findings align with this. However, Graf and Kinshuk (2009) argue that personalization based solely on learning styles is insufficient, highlighting the need to include motivation and contextual factors.

A preliminary pilot study has not yet been conducted. Instead, the current phase of the project focused on the design

and technical development of the adaptive system. Empirical testing with real classroom data is planned for future research, subject to ethical approval and informed consent procedures.

International Contribution

In addition to its relevance for the Turkish context, this study is expected to provide insights into the broader discussion on adaptive learning. The redesigned modules on an open-source LMS may serve as an example of how adaptive personalization could be applied in different cultural and pedagogical settings, potentially offering a useful point of reference for future studies.

Contributions to Educational Technology

This system provides an original contribution to the field of personalized e-learning in Turkiye. While existing systems generally deliver standardized content, this system dynamically modifies content flow in real time through learner profiling algorithms.

Limitations

The system has not yet been tested with a large-scale learner sample, which limits the generalizability of findings. Additionally, the focus was on system development, not on empirical measurement of learning outcomes. These constraints should be addressed in future studies.

CONCLUSION

This study aimed to develop an adaptive e-learning system personalized according to learning styles. The system enhances the learning experience by considering individual differences among students, making it more meaningful and effective. Through learner profiling, content adaptation, and feedback mechanisms, the system enables each student to experience a unique learning journey.

- The system demonstrates that students with different learning profiles can reach the same learning objectives through different pathways. This supports the applicability of the learner-centered approach in digital environments.
- Data security is a critical factor in adaptive systems. The secure storage and anonymization of data collected during learner profiling will become increasingly important in the future.

Recommendations

- **Motivation and context:** Future studies should include not only learning styles but also motivation, learning context, and emotional states in the profiling process.
- **AI-powered recommendation engines:** The system can be enhanced with AI-based recommendation engines to provide learners with more flexible learning pathways.
- **Semi-automatic content generation:** Modules that support instructors in content development may reduce workload.
- Long-term studies should measure the impact of this system on academic achievement, student satisfaction, and learning retention.

In conclusion, this system makes a significant contribution to educational technologies in terms of personalization and adaptability. Such systems, which support learner-centered approaches in the process of digital transformation in education, are expected to become widely adopted in the future.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Altay, M. (2024). Öğrenme kavramı ve bireysel farklılıklar. Akademi Yayıncılık.

Brusilovsky, P., & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In the adaptive web: methods and strategies of web personalization (pp. 3-53). Berlin, Heidelberg: Springer Berlin Heidelberg.

Davis, N., Little, P., & Stewart, S. (2021). Digital transformation in education: adaptive systems and personalized learning. *J Edu Techno Res, 42*(2), 115-132. doi:10.1016/j.edutech.2021.03.005

Graf, S., & Kinshuk, K. (2009). Advanced adaptivity in learning management systems: model-driven approaches. *Int J Learn Technol, 5*(3), 256-274. doi:10.1504/IJLT.2009.028804

Janssen, J. (2021). The role of IT infrastructure in adaptive learning ecosystems. *Comput Edu, 175*, 104329. doi:10.1016/j.compedu.2021.104329

McKenney, S., & Reeves, T. C. (2020). Conducting educational design research (2nd ed.). Routledge. doi:10.4324/9780429203374

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: an updated survey. *Wiley Interdiscipl Rev Data Mining Knowl Discov, 10*(3), e1355. doi:10.1002/widm.1355

Ruman, H. (2022). Personalized learning and adaptive systems in the digital age. *Int J Instruct Design, 15*(1), 44-63.

Süral, İ. (2021). Açık kaynak tabanlı öğrenme yönetim sistemlerinin adaptif kullanımı. *Eğit Teknol Derg, 12*(4), 88-104.

Tuna, G. (2020). Adaptif öğrenme ortamlarının metodolojik yaklaşımları. *Eğit Araşt Derg, 18*(2), 56-72.