# EMERGING ARTIFICIAL INTELLIGENCE APPLICATIONS IN COMPUTING, EDUCATION, AND INNOVATION

EDITOR
PROF. DR. EYYÜP GÜLBANDILAR




BİDGE Yayınları

**BİDGE Yayınları**

**Emerging Artificial Intelligence Applications in Computing, Education, and Innovation**

**Editor:** Prof. Dr. EYYÜP GÜLBANDILAR

**ISBN:** 978-625-5057-96-9

1st Edition
Page Layout By: Gözde YÜCEL
Publication Date: 2026-10-25
BİDGE Yayınları

All rights reserved. No part of this work may be reproduced in any form or by any means, except for brief quotations for promotional purposes with proper source attribution, without the written permission of the publisher and the editor.

Certificate No: 71374
All rights reserved © BİDGE Yayınları
www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com
Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.
Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya / Ankara

## CONTENTS

ANALOG IN-MEMORY COMPUTING ACCELERATORS: ARCHITECTURE, DATAFLOW, AND THE SIMULATION TOOLCHAIN ..... 1

*BAKİ GÖKGÖZ*

AN ALGORITHM FOR AUTOMATIC DICTIONARY USING THE WEB ..... 25

*AHMET TOPRAK, METİN TURAN*

ARTIFICIAL INTELLIGENCE ENHANCED CYBERSECURITY IN DIGITAL EDUCATION ..... 59

*ÖMER SEVİNÇ*

THE IMPACT OF TECHNOPARKS ON FIRM PERFORMANCE: AN EMPIRICAL INVESTIGATION INTO PRODUCTIVITY, INNOVATION, RECYCLING, EMPLOYMENT, AND EXPORTS ..... 68

*AYHAN AYDIN*

# CHAPTER 1

# Analog In-Memory Computing Accelerators: Architecture, Dataflow, and the Simulation Toolchain

**Baki GÖKGÖZ**

Asst. Prof., Department of Computer Technologies, Torul Vocational School, Gümüşhane University, Gümüşhane, Türkiye. ORCID: 0000-0001-8166-6282. E-mail: bakigokgoz@gumushane.edu.tr

**Abstract**

Energy-efficient inference has become a central concern in deep learning, and analog in-memory computing (AIMC) is among the architectures proposed to address it. The approach targets the memory wall, the bottleneck that limits conventional von Neumann machines, by performing multiply–accumulate operations inside dense arrays of memory devices rather than moving weights to a separate processing unit. This chapter serves as an architecture-level introduction to how these accelerators are built and how they are evaluated before fabrication. The crossbar primitive and its surrounding circuits are described first, followed by the mapping of neural-network layers onto a grid of tiles and the dataflow decisions that govern scheduling across the chip. The open simulation tools used to reason about accuracy, energy, and area are then reviewed, along with the assumptions that determine how far their results can

be trusted. No specific design is advocated and no new measurements are presented; the aim is a coherent overview of the field and the vocabulary required to read its literature.

Keywords: analog in-memory computing; crossbar accelerator; dataflow; hardware simulation; design-space exploration; deep-learning inference.

## Introduction

Where data is stored, rather than how quickly it is multiplied, sets the real cost in contemporary deep-learning hardware (Horowitz, 2014; Sze et al., 2017). A network keeps its weights in memory, and a conventional processor must move each weight across a bus to an arithmetic unit for every multiply–accumulate operation, then return the result. For a model with millions of parameters, the energy spent on this traffic exceeds that of the computation by a wide margin (Sze et al., 2017). The underlying problem is a long-standing mismatch between fast arithmetic and comparatively slow memory access, commonly termed the memory wall. Circumventing it is precisely what makes architectures that compute inside memory, instead of alongside it, worth pursuing (Figure 1) (Ielmini & Wong, 2018; Sebastian et al., 2020).

Analog in-memory computing follows that path (Hu et al., 2018; Sebastian et al., 2020). Its principle can be stated compactly: memory devices are placed on a grid, and the electrical behaviour of the grid itself carries out the arithmetic. If voltages drive the rows and each device is set to a chosen conductance, then the current collecting on a given column emerges as the weighted sum of the inputs, a direct consequence of Ohm's and Kirchhoff's laws. One read of the array thus corresponds to one complete matrix–vector product. Because the weights are never relocated, the energy formerly spent on movement is largely removed, and the columns are resolved concurrently rather than sequentially.

A simple estimate conveys the magnitude of the saving. Suppose a weight matrix contains a million entries. A conventional machine retrieves a million weights, carries out a million multiplications, accumulates the results, and incurs a data-movement cost at each retrieval. A crossbar that stores the same million weights reaches the result in only a few analog steps, holding the weights in place and letting the wires perform the accumulation. The many cycles and memory accesses of a digital machine are compressed into a single array read. This compression, not any gain in the speed of one multiplication, is where the advantage originates, which also explains why the method helps most in the large, weight-dense layers in which data movement is especially expensive.

The purpose of this chapter is to explain how accelerators are organized around that primitive and how they are investigated. The treatment is intentionally broad. Device physics is not derived, no new design is put forward, and no experiments are reported. The intended outcome is a working understanding of three things: the constituents of an AIMC accelerator, the manner in which a network is mapped onto it, and the simulation tools the field employs to anticipate behaviour ahead of fabrication. Equipped with that understanding, a reader will find the diagrams and claims of the primary literature considerably more approachable.

It is useful to mark one contrast early, because it accounts for both the appeal of the approach and its chief difficulty. Digital accelerators, whether general-purpose GPUs or dedicated tensor engines, still keep storage and computation apart. Their efficiency derives from situating numerous arithmetic units close to fast on-chip memory and from reusing operands wherever possible, yet each weight is still fetched and each product is still produced by a digital multiplier. AIMC breaks with this arrangement more decisively: multiplication is carried out by the device instead of a logic gate, and summation by a wire instead of an adder tree. Thousands of

multiply–accumulate operations can run on a single array with almost no data movement and no explicit arithmetic circuitry. The accompanying difficulty is that analog computation is inherently approximate, and a substantial part of the engineering covered in this chapter is devoted to containing that approximation without forfeiting the efficiency that justified the approach to begin with.

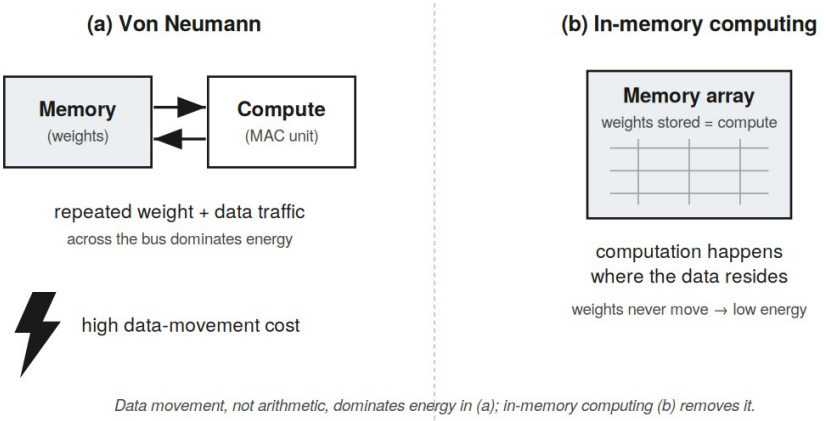



*Figure 1. Data movement, not arithmetic, dominates energy in a conventional von Neumann machine (a), where weights and activations repeatedly traverse the bus between memory and the compute unit. In-memory computing (b) performs the computation where the data resides, so the weights never move.*

The chapter is structured as follows. The crossbar and its peripheral circuits are introduced in Section 2. Section 3 explains how a network's layers are partitioned and placed on a grid of tiles. Dataflow and scheduling are taken up in Section 4. Section 5 reviews the simulation tools and the quantities they estimate, and Section 6 examines the assumptions that limit the reliability of those estimates. Section 7 surveys the directions the field is taking, and Section 8 offers a conclusion. Throughout, the discussion remains at the level of architecture and method, and presumes only that the reader is

familiar with neural networks and with the matrix multiplication that underlies most of their computation.

## The Crossbar Primitive and Its Periphery

### The array

An AIMC accelerator is built upon the crossbar array (Figure 2). Horizontal wires, the word lines or rows, cross vertical wires, the bit lines or columns, and a programmable memory device occupies each intersection. The value held by a device is physical, in most cases an electrical conductance. The role can be filled by a variety of device technologies, among them resistive RAM, phase-change memory, floating-gate transistors, and the more recent ferroelectric and magnetic devices (Ielmini & Wong, 2018; Yu, 2018). Throughout this chapter the device is treated as a black box, namely a programmable and roughly linear resistor whose conductance represents a weight.

Suppose the rows are driven by input signals while the columns are held at a shared potential. Every device then conducts a current set by the product of its input and its conductance, and these currents combine on the column wire. What appears on a column is consequently the dot product between the input vector and the conductances stored in that column. Carrying this out across all columns simultaneously realizes a complete matrix–vector multiplication in one analog step, and that is where the efficiency of the array originates (Hu et al., 2018; Yao et al., 2020).

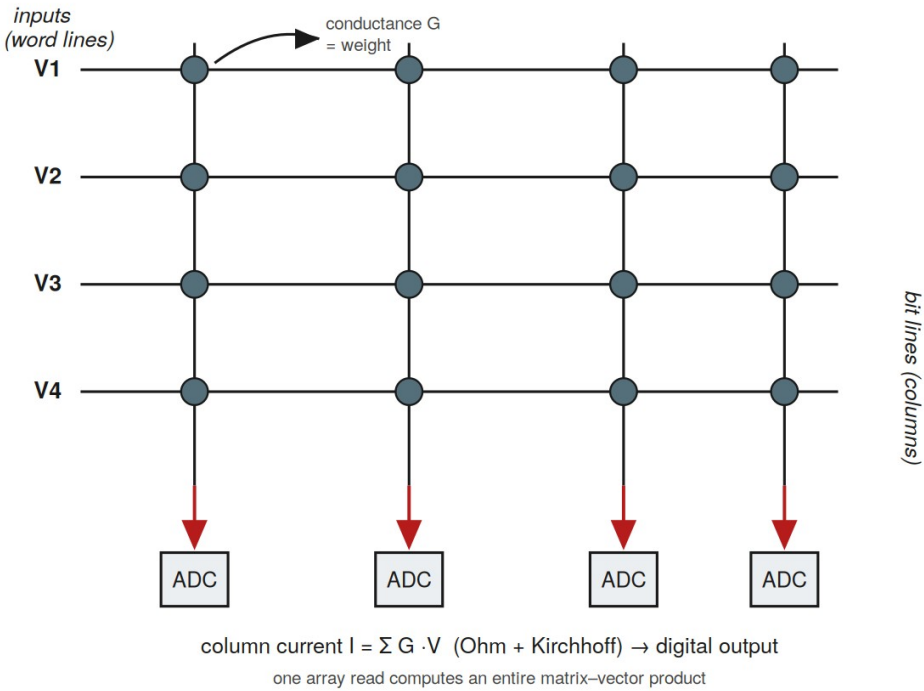



*Figure 2. The crossbar primitive. Input signals drive the horizontal word lines; a programmable device at each intersection stores a conductance that encodes a weight. The currents contributed by all devices on a column sum along the vertical bit line, and the column current, by Ohm's law and Kirchhoff's current law, is the dot product of the inputs with that column's conductances. A single array read therefore computes an entire matrix–vector product, which is digitized by the column converters.*

Two practical issues surface at once. A conductance cannot take a negative value, but a weight can. The standard remedy represents each weight as the difference between two conductances, realized with a column pair or with a reference column whose current is subtracted (Chi et al., 2016; Shafiee et al., 2016). The second issue is limited dynamic range: because one device stores little precision, a

high-precision weight is spread across several devices or columns and reassembled with suitable scaling. Although invisible at the algorithm level, these encoding choices have a pronounced effect on the area and energy of the array.

A third consideration belongs to the input side rather than the weight side. The signals that drive the rows must represent the activations entering the layer, and their fidelity places a ceiling on the fidelity of the entire computation. Supplying activations at full precision in a single step makes the input circuitry costly, whereas supplying them bit by bit across several cycles simplifies the circuitry at the expense of speed. The array is unaffected by this decision, but the design surrounding it is not, and the resulting tension between input precision and speed is a recurring feature of AIMC architecture.

The economy of the analog summation is also worth appreciating. In a digital accelerator the products along a column are added by a tree of adders whose depth increases with the number of terms. A crossbar instead performs the summation in one stroke on the shared column wire, which collects the current from every device together. A sum over hundreds or thousands of terms is thereby completed in a single electrical event, without an adder tree and without intermediate storage. This is the crux of the approach, and it is the reason so much design effort is directed at preventing the surrounding circuitry from dissipating the advantage the array supplies.

**Peripheral circuits**

On its own, an array of conductances achieves nothing. Circuits are needed to drive it and to read its results, and a large share of an accelerator's complexity and cost resides in that periphery.

At the input, the digital activations entering a layer have to be turned into signals the array will accept. This is frequently done by

encoding each input as a train of pulses, or as a sequence of bits delivered over several cycles, an arrangement known as bit-serial input. Such encoding exchanges time for simplicity, since greater input precision calls for more cycles per operation.

At the output, each column yields an analog current that must become a digital number before it can be stored, accumulated across tiles, or forwarded to the next layer. The conversion is carried out by an analog-to-digital converter placed on the column (Ankit et al., 2019; Shafiee et al., 2016). It is one of several peripheral components, which also include sample-and-hold circuits, current-to-voltage conversion, and the shift-and-add logic that reassembles bit-serial and multi-column partial results. The salient point is that, taken together, the periphery accounts for a large portion of the chip's area and energy and remains an active subject of circuit research. These components are treated here by what they do rather than by their transistor-level realization.

Sharing the periphery across many rows distributes its cost over many operations, which is part of the reason AIMC can remain efficient in spite of the conversion overhead. The benefit of sharing is bounded, however. A converter that seems inexpensive when amortized over a wide array can come to dominate once arrays become small, so the balance between array size and peripheral cost is a persistent concern. Wide arrays amortize the periphery better but are more difficult to drive accurately and more susceptible to the parasitic resistance of long wires; narrow arrays are electrically cleaner yet pay a heavier peripheral cost per operation. The architecture is therefore a compromise between the analog core and the circuits that render it usable, and the two cannot be designed independently. This interdependence reappears below, in the treatment of system-level evaluation.

**From array to tile**

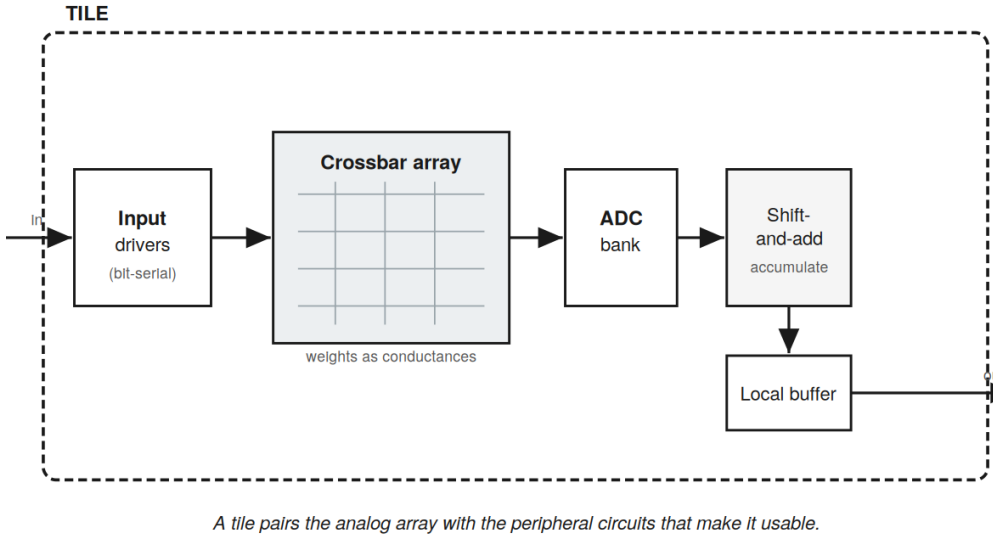



*Figure 3. Anatomy of a tile. The crossbar array is wrapped in the peripheral circuits that make it usable: input drivers that present activations to the rows, a bank of converters that digitize the column currents, shift-and-add logic that recombines bit-serial and multi-column partial results, and a local buffer. A tile is the self-contained unit from which larger accelerators are built.*

A single array is small. The weight matrices of real networks far exceed what one array can store, and in practice fabrication and signal-integrity constraints hold array dimensions down in any case. For that reason the practical unit of design is the tile rather than the bare array (Figure 3): an array accompanied by its input drivers, its column converters, its local accumulation logic, and a small buffer. When assigned a portion of a weight matrix, a tile computes the corresponding matrix–vector product independently. Larger accelerators are assembled by replicating tiles and linking them with an on-chip network (Ankit et al., 2019; Song et al., 2017), which brings up the subsequent question of how an entire network is distributed across many tiles.

## Mapping Neural Networks onto the Fabric

### Partitioning weights

At bottom, a neural-network layer is a matrix of weights. A fully connected layer is a matrix already, and a convolutional layer becomes one after its filters are unrolled so that the convolution is cast as a matrix multiplication (Ankit et al., 2019; Chen, Emer, & Sze, 2016). To map a layer onto the hardware is thus to place that matrix into the physical arrays (Figure 4).

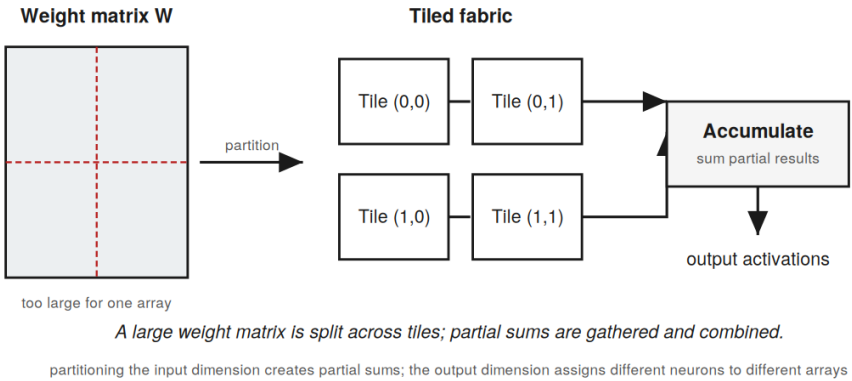



*Figure 4. Mapping a layer onto the fabric. A weight matrix too large for one array is partitioned across several tiles. Partitioning the input dimension means each tile computes a partial sum over a subset of the inputs, and these partial sums are gathered and accumulated downstream; partitioning the output dimension assigns different output neurons to different tiles.*

A matrix larger than a single array is divided into tiles along both of its dimensions. Division along the input dimension leaves no single array with a view of the entire input vector; each array forms a partial sum over the inputs it contains, and those partial sums are

combined downstream. Division along the output dimension is more straightforward, since it merely sends different output neurons to different arrays. In both cases the accelerator must route inputs to the appropriate arrays and then collect the partial results, a bookkeeping duty assigned to the on-chip interconnect and the accumulation logic.

**Layer-to-tile assignment**

Dividing one matrix addresses only part of the task. The designer must further decide how the layers of a complete network share the fabric, and two broad strategies present themselves. Under a weight-stationary layout, each layer is granted dedicated tiles that are programmed once and retained for the whole workload (Ankit et al., 2019; Song et al., 2017). This yields high throughput, since the layers can run concurrently in a pipeline, but it requires sufficient hardware to accommodate the entire network at the same time. Under a time-multiplexed layout, a smaller set of tiles is reprogrammed to host one layer after another. Area is reclaimed in exchange for the time and energy of rewriting weights, an arrangement that suits cases where devices are cheap to program and the workload can absorb the extra latency.

The choice ranks among the weightier decisions in AIMC design, and it is strongly conditioned by the device. Programming that is slow or expensive favours weight-stationary layouts, while fast and endurant devices make time-multiplexing viable. Rather than resolve the trade-off, this chapter marks it as one of the principal axes along which designs diverge.

The decision additionally shapes the workloads a design accommodates. An accelerator that holds a complete network in a weight-stationary fashion suits a fixed inference task applied to a stream of inputs, such as continuous classification at the edge, where the weights are programmed once and left unchanged. A time-

multiplexed design is preferable where many models must share a constrained hardware budget, or where one piece of hardware serves distinct tasks at distinct times. Neither dominates universally; each encodes an assumption about how the hardware will be used, and discerning that assumption is frequently the fastest route to a design's priorities.

**The heterogeneity of a mapped network**

A point that is easy to overlook yet matters a great deal is that a mapped network is not uniform. Its layers vary in size, in the statistics of their activations, in their location within the network, and in the number of arrays they require. A layout well suited to one layer can prove wasteful for another. A good deal of the refinement in AIMC architecture consists of acknowledging this unevenness and apportioning resources to match it, instead of handling every layer the same way. Here it is the principle that counts; the detailed per-layer allocation policies are the province of specialized design studies.

The unevenness extends beyond mere counts. The early convolutional layers operate on large spatial maps with few channels, whereas the later layers operate on small maps with many channels; the former generate numerous small matrix multiplications and the latter a smaller number of larger ones, with the fully connected layers at the end presenting yet another profile. Owing to these differences, the arrays hosting different layers experience markedly different utilization, so that a fabric scaled to the heaviest layer may lie idle while hosting the lightest. Maintaining balanced utilization, so that costly hardware is not left unused, is one of the less conspicuous challenges in AIMC, and it comes into view only once a network is regarded as something other than a uniform stack of identical operations.

**Dataflow and Scheduling**

## Why dataflow matters

With the weights resident on tiles, the accelerator still has to move activations through them in a particular order. The term dataflow denotes how computation and data transfer are scheduled across the chip in space and time (Chen, Krishna, et al., 2017; Chen, Emer, & Sze, 2016). A single set of tiles can operate under markedly different dataflows, and the choice bears on latency, on the buffer capacity required, and on the energy consumed in relaying intermediate results.

Dataflow is readily undervalued, concerned as it is with scheduling rather than arithmetic, yet it frequently decides whether an accelerator attains its efficiency targets. Two designs with identical arrays and identical peripheral circuits may nonetheless diverge considerably in throughput and energy, for no reason other than that one schedules its work more adeptly. The arrays constitute a fixed resource; the dataflow dictates how thoroughly they are exploited and how much supplementary movement is required to keep them supplied. In this sense dataflow stands to an AIMC accelerator as an instruction schedule stands to a processor: absent from the block diagram, but pervasive in the measured outcome.

## Pipelining across layers

Since a weight-stationary layout holds every layer at once, it lends itself naturally to pipelining. The moment the first layer completes one input, it may accept the next while the second layer processes the output of the first. In the steady state every layer is occupied with a different sample, and throughput stays high. The attendant cost is buffering: the output of each stage must be retained until the following stage is prepared to consume it, and any imbalance among stages produces bottlenecks in which faster layers wait on slower ones.

## Spatial and temporal reuse

Reuse, too, is governed by the dataflow. Within a convolutional network a single filter is applied across many spatial positions, and a single input patch contributes to many outputs. A judicious dataflow capitalizes on this by keeping one operand stationary and streaming the other past it, so that no value travels more than it has to. The dataflow categories familiar from digital accelerators, among them the weight-stationary, output-stationary, and row-stationary schemes (Chen, Krishna, et al., 2017; Chen, Emer, & Sze, 2016), transfer to the analog setting, subject to the important proviso that AIMC fixes the weights physically within the array and so restricts the available options. Grasping a given accelerator often amounts to recognizing which operand it holds stationary and which it streams.

### The role of the interconnect

The partial sums generated by different tiles have to be gathered and combined, and activations have to be delivered to the tiles requiring them. The on-chip interconnect responsible for this routing is a first-class element of the design, not an accessory appended at the end. Its topology and bandwidth decide whether the arrays, fast as they are, spend their time computing or idling for want of data. A lesson the field encounters repeatedly is that the analog core may be rendered extremely efficient only for the surrounding digital and communication logic to command the overall energy and area (Shafiee et al., 2016; Song et al., 2017). An accelerator must accordingly be assessed as a complete system rather than as an array considered in isolation.

## The Simulation Toolchain

### Why simulate

Because fabricating a chip is slow and costly, the great majority of questions about AIMC can be settled well in advance of

tape-out. Simulation is consequently the field's principal means of predicting how a design will behave (Peng et al., 2019; Rasch et al., 2021; Xiao et al., 2022). A capable simulator enables a researcher to ask what accuracy a network would attain on a particular hardware configuration, what that configuration would cost in energy and area, and how these figures move as the design parameters are swept. Since such tools lie behind almost every published result, familiarity with them is nearly as important as familiarity with the hardware.

**Levels of modelling**

Simulators occupy a spectrum of fidelity. At one extreme are functional models, which capture only the mathematical effect of the hardware by injecting quantization and noise into a network's computations to estimate the accuracy impact, with no circuit modelled at all. At the other extreme are circuit-level models, which estimate energy, latency, and area by accounting for the arrays, the converters, the buffers, and the interconnect, typically through a calibrated model of each component (Dong et al., 2012; Peng et al., 2019). Occupying the middle ground are behavioural models, which approximate device and circuit non-idealities in sufficient detail to study their effect on accuracy while remaining fast enough to run a complete network (Joshi et al., 2020; Rasch et al., 2021).

In practice a workflow draws on several levels. A functional model resolves accuracy questions quickly during exploration, and a circuit-level model then furnishes the energy and area figures that substantiate an efficiency claim. Both are necessary, for neither on its own conveys the full merit of an accelerator.

The level of modelling likewise governs how far a result may be generalized. A functional model that injects an abstract noise term can reproduce the qualitative character of an effect while remaining indifferent to the particular circuit that gives rise to it, which renders its conclusions broad but coarse. A circuit-level model bound to one

technology produces sharper figures that may not carry over to another technology. A careful study is explicit about the level at which it operates, which is tantamount to being explicit about the reach of its claims. For the reader, pinning down that level is the initial step in judging how broadly a result applies.

**Representative open frameworks**

A number of open frameworks have emerged as shared reference points that any reader of the AIMC literature meets repeatedly. One widely adopted family offers hierarchical, device-to-system estimation coupled to a deep-learning front end, so that a network expressed in a standard framework can be assessed for accuracy together with hardware cost; successive releases have steadily refined the modelling of device- and circuit-level non-idealities (Peng et al., 2019; Peng et al., 2021). Other frameworks concentrate on faithfully reproducing the analog behaviour of crossbars for accuracy studies (Xiao et al., 2022), while still others integrate hardware-aware training so that a network may be adapted to withstand analog imperfections (Joshi et al., 2020; Rasch et al., 2021). These tools differ in their assumptions and in the aspects each models most carefully, and using them well depends on knowing which questions each was constructed to address.

**What the tools report**

Across frameworks, the reported quantities are largely shared. On the accuracy side, a simulator states how a network performs once hardware effects are incorporated, a figure that can lie below the clean software accuracy. On the efficiency side, the tools give energy per operation or per inference, chip area, and latency or throughput, usually itemized by component so that the designer can discern whether the arrays or the periphery dominate. Such itemized breakdowns are often more informative than the headline figure,

since they expose where the cost truly accrues and hence where effort ought to be concentrated.

### Simulation as design-space exploration

A simulator is at its most useful when it sweeps a range of possibilities instead of scoring one design. Seldom is the performance of a single configuration interesting in itself; the instructive question is how performance shifts as a parameter is varied, since that is what a design decision hinges on. By running many configurations and setting their outcomes side by side, a simulator converts an isolated measurement into a map of the design space, on which favourable regions can be sought and unfavourable ones avoided. Here, more than anywhere, speed is decisive: a tool able to evaluate hundreds of configurations affords a breadth of comparison that fabrication could never match. A caveat applies, though, in that the swept configurations must be kept comparable, so that any difference in outcome is attributable to the parameter under study and not to some incidental change elsewhere in the setup.

## Modelling Assumptions and Their Limits

### Every estimate rests on assumptions

A simulation result is reliable only to the degree that the assumptions beneath it are. This is not a feature peculiar to AIMC but a property of modelling in general. In AIMC, however, the assumptions are both numerous and consequential, and they are easily missed when a tool returns a single clean number. The prudent course is to ask what was assumed before accepting what was reported.

### Common assumptions

A handful of assumptions recur. Device behaviour is idealized to one degree or another: actual devices differ from one another, drift with time, and respond non-linearly, and a simulator

reflects this only insofar as its models extend (Joshi et al., 2020; Yu, 2018). The periphery is represented by calibrated estimates whose accuracy is contingent on the target technology. The network mapping, the dataflow, and the input and output precision are all chosen by the user, and two equally reasonable sets of choices can yield materially different conclusions. Even the selection of which test data to evaluate, and how much of it, influences the reported accuracy.

**Reading results critically**

AIMC results are best construed, then, as conditional statements: a stated accuracy or efficiency holds under one particular set of assumptions and configurations. Two studies that seem to conflict may merely have adopted different, individually defensible choices. For this reason the methods sections of AIMC papers warrant careful reading, and reproducibility, the release of the configurations and data behind a result, has grown into an increasingly standard expectation (Gholami et al., 2022; Peng et al., 2021). For a newcomer, the most serviceable habit is to regard every headline number as the terminus of a chain of assumptions and to follow that chain back before reaching any conclusion.

**Trends and Open Directions**

A number of directions are shaping the near-term development of AIMC architecture. The first is the ongoing transfer of attention from the analog core to the digital periphery and interconnect, as designers recognize that system-level efficiency is capped by the least efficient stage of the pipeline. The second is a growing reliance on co-design, in which training, mapping, and hardware are optimized jointly rather than in isolation, so that the network is tailored to the hardware and the hardware to the network (Rasch et al., 2023; Rasch et al., 2021). The third is the maturation of the simulation toolchain itself, including work to validate simulators

against measured silicon and to establish where different tools concur and where they part, which reinforces the basis on which design decisions rest (Wan et al., 2022; Xiao et al., 2022).

A fourth direction concerns the range of workloads AIMC is expected to support. Much of the early work centred on image classification with compact convolutional networks, in part because they are small enough to map completely and in part because they furnish convenient benchmarks. Carrying the approach to larger and differently structured models, such as those used for sequence processing and generative tasks, raises fresh questions: how to map operations that do not reduce as neatly to fixed weight matrices, and how to schedule computation when the reuse pattern departs from that of a convolution. A fifth direction is integration. An AIMC accelerator seldom works in isolation, and the manner in which its tiles cooperate with conventional digital compute, with memory hierarchies, and with the remainder of a system-on-chip is becoming part of the design problem rather than a later concern.

None of these questions is settled, and each constitutes an opening for further work. The architecture-level viewpoint advanced in this chapter is meant to place the reader in a position to follow these conversations, and in due course to take part in them.

## Conclusion

Analog in-memory computing converts a grid of programmable memory devices into a parallel engine for matrix–vector multiplication, and by computing where the data resides it holds out the prospect of large energy savings in deep-learning inference. Realizing that prospect is an exercise in architecture no less than in device physics. The crossbar must be surrounded by input and output circuitry, weight matrices must be partitioned and spread across many tiles, and activations must be scheduled through the fabric by a dataflow that respects the fixed placement of the

weights. And because fabrication is expensive, the field conducts its reasoning through open simulation tools that estimate accuracy, energy, and area, with confidence in those estimates resting on assumptions that merit scrutiny. The chapter has endeavoured to draw these elements together into a single picture. A reader who retains the anatomy of a tile, the logic of mapping and dataflow, and a measured wariness toward figures quoted without their conditions will be well positioned to read the primary literature and to judge where the outstanding opportunities lie.

**Kaynakça/References**

Ankit, A., Hajj, I. E., Chalamalasetti, S. R., Ndu, G., Foltin, M., Williams, R. S., Faraboschi, P., Hwu, W. W., Strachan, J. P., Roy, K., & Milojicic, D. S. (2019). PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 715–731). ACM. https://doi.org/10.1145/3297858.3304049

Chen, Y.-H., Emer, J., & Sze, V. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In Proceedings of the 43rd International Symposium on Computer Architecture (pp. 367–379). IEEE. [DOI doğrulanamadı (kontrol edin)]

Chen, Y.-H., Krishna, T., Emer, J. S., & Sze, V. (2017). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits, 52(1), 127–138. https://doi.org/10.1109/JSSC.2016.2616357 [DOI doğrulanamadı (kontrol edin)]

Chi, P., Li, S., Xu, C., Zhang, T., Zhao, J., Liu, Y., Wang, Y., & Xie, Y. (2016). PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In

Proceedings of the 43rd International Symposium on Computer Architecture (pp. 27–39). IEEE. [DOI doğrulanamadı (kontrol edin)]

Dong, X., Xu, C., Xie, Y., & Jouppi, N. P. (2012). NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31(7), 994–1007. https://doi.org/10.1109/TCAD.2012.2185930 [DOI doğrulanamadı (kontrol edin)]

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2022). A survey of quantization methods for efficient neural network inference. In Low-power computer vision (pp. 291–326). Chapman and Hall/CRC. https://arxiv.org/abs/2103.13630

Horowitz, M. (2014). 1.1 Computing's energy problem (and what we can do about it). In IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers (pp. 10–14). IEEE. https://doi.org/10.1109/ISSCC.2014.6757323 [DOI doğrulanamadı (kontrol edin)]

Hu, M., Graves, C. E., Li, C., Li, Y., Ge, N., Montgomery, E., Davila, N., Jiang, H., Williams, R. S., Yang, J. J., Xia, Q., & Strachan, J. P. (2018). Memristor-based analog computation and neural network classification with a dot product engine. Advanced Materials, 30(9), 1705914. https://doi.org/10.1002/adma.201705914

Ielmini, D., & Wong, H.-S. P. (2018). In-memory computing with resistive switching devices. Nature Electronics, 1(6), 333–343. https://doi.org/10.1038/s41928-018-0092-2

Joshi, V., Le Gallo, M., Haefeli, S., Boybat, I., Nandakumar, S. R., Piveteau, C., Dazzi, M., Rajendran, B., Sebastian, A., & Eleftheriou, E. (2020). Accurate deep neural network inference using computational phase-change memory. Nature Communications, 11, 2473. https://doi.org/10.1038/s41467-020-16108-9

Peng, X., Huang, S., Jiang, H., Lu, A., & Yu, S. (2021). DNN+NeuroSim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(11), 2306–2319. https://doi.org/10.1109/TCAD.2020.3043731

Peng, X., Huang, S., Luo, Y., Sun, X., & Yu, S. (2019). DNN+NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies. In IEEE International Electron Devices Meeting (IEDM) (pp. 32.5.1–32.5.4). IEEE. https://doi.org/10.1109/IEDM19573.2019.8993491 [DOI doğrulanamadı (kontrol edin)]

Rasch, M. J., Moreda, D., Gokmen, T., Le Gallo, M., Carta, F., Goldberg, C., El Maghraoui, K., Sebastian, A., & Narayanan, V. (2021). A flexible and fast PyTorch toolkit for simulating training and inference on analog crossbar arrays. In IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS) (pp. 1–4). IEEE. https://doi.org/10.1109/AICAS51828.2021.9458494

Rasch, M. J., Mackin, C., Le Gallo, M., Chen, A., Fasoli, A., Odermatt, F., Li, N., Nandakumar, S. R., Narayanan, P., Tsai, H., Burr, G. W., Sebastian, A., & Narayanan, V. (2023). Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators. Nature Communications, 14, 5282. https://doi.org/10.1038/s41467-023-40770-4

Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., & Eleftheriou, E. (2020). Memory devices and applications for in-memory computing. Nature Nanotechnology, 15(7), 529–544. https://doi.org/10.1038/s41565-020-0655-z

Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J. P., Hu, M., Williams, R. S., & Srikumar, V. (2016). ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In Proceedings of the 43rd International Symposium on Computer Architecture (pp. 14–26). IEEE. https://doi.org/10.1145/3007787.3001139

Song, L., Qian, X., Li, H., & Chen, Y. (2017). PipeLayer: A pipelined ReRAM-based accelerator for deep learning. In IEEE International Symposium on High Performance Computer Architecture (HPCA) (pp. 541–552). IEEE. https://doi.org/10.1109/HPCA.2017.55

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Wan, W., Kubendran, R., Schaefer, C., Eryilmaz, S. B., Zhang, W., Wu, D., Deiss, S., Raina, P., Qian, H., Gao, B., Joshi, S., Wu, H., Wong, H.-S. P., & Cauwenberghs, G. (2022). A compute-in-memory chip based on resistive random-access memory. Nature, 608(7923), 504–512. https://doi.org/10.1038/s41586-022-04992-8

Xiao, T. P., Feinberg, B., Bennett, C. H., Agarwal, S., Jacobs-Gedrim, R., Fuller, S. J., Marinella, M. J., & Agarwal, S. (2022). An accurate, error-tolerant, and energy-efficient neural network inference engine based on SONOS analog memory. IEEE Transactions on Circuits and Systems I: Regular Papers, 69(4), 1480–1493. https://doi.org/10.1109/TCSI.2021.3134313

Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., Yang, J. J., & Qian, H. (2020). Fully hardware-implemented memristor convolutional neural network. Nature, 577(7792), 641–646. https://doi.org/10.1038/s41586-020-1942-4

Yu, S. (2018). Neuro-inspired computing with emerging nonvolatile memory. Proceedings of the IEEE, 106(2), 260–285. https://doi.org/10.1109/JPROC.2018.2790840

# CHAPTER 2

# An Algorithm for Automatic Dictionary Using the Web

**AHMET TOPRAK**[1]
**METİN TURAN**[2]

## 1. Introduction

There has been an increasing need to extract information from big data lately. Domain-specific dictionaries are needed to manage this task. Therefore, studies in language lexicography have been focusing on automatic dictionary creation. These dictionaries can be used for search engine optimization [39], automatic summarizing systems [19, 15], theme determination [10] and text classification [28] researches.

The dictionary creation process can be done manually [24], semi-automatically [25], or automatically [34]. Hand-created dictionaries are static, and they require continuous external support to increase the size of the dictionary, resulting in higher maintenance costs. Riloff [34] described the shortcomings of manual dictionaries.

[1] Asst. Prof., İstanbul Gelisim University, Computer Engineering, Orcid: 0000-0001-7046-8512
[2] Assoc. Prof., İstanbul Ticaret University, Computer Engineering, Orcid: 0000-0002-1941-6693

Knowledge-based natural language processing (NLP) [27] systems are often criticized although they have achieved success with specific tasks because they are dependent on a domain-specific dictionary that requires manual engineering knowledge. Because these systems cannot be easily scaled or transferred to the new fields.

In this study, an algorithm was proposed and experimentally tested to create an automatic dictionary for a specific domain. The initial step is to find meaningful words from the document (or set of documents) given by the user. This document is called "reference document". The most meaningful words in the reference document were identified using term frequency-inverse document (TF–IDF) values. The first dictionary words, we call it "seed words", are obtained from the most meaningful words of the reference document. A web search is then applied using the Azure Web Cognitive Search Interface repetitively in order to execute a new query and find new related words within returned documents. These meaningful words found at each iteration aren't added directly to the dictionary. Instead, the WordNet dictionary is used to evaluate the similarity of each meaningful word within the dictionary so that adding meaningless words to the dictionary is prevented. The words with higher similarity values above a certain threshold value (experimentally learned) are added to the dictionary until the desired number of words in the dictionary is accumulated. Hence, the dictionary continues growing automatically and continuously without any intervention from human being.

The hash similarity metric was used to evaluate the overall similarity of the dictionary. The notion of locality-sensitive hashing was first introduced by Indyk et al. [20]. For efficient near-neighbor search, the locality-sensitive hashing (LSH) technique exploits special hash functions that make buckets contain similar keys (data), yet it does not guarantee that all data in a bucket are similar to each other. The so-called locality-sensitive hash functions provide a high

probability for similar data to be in the same bucket but a low probability for dissimilar data to be in the same bucket [26].

The following section includes prior works in the field of dictionary generation. The methods and studies used for dictionary creation are explained in the third section. The tests applied within the research are explained in the fourth section. Finally, the results are discussed and evaluated in the last section.

## 2. Literature Review

TF–IDF has been successfully applied to many different NLP applications. One of these important studies was by Paik [29], who performed an effective ranking study in 2013 using TF–IDF. In this particular study, Paik addressed the limitations of the pivoted length normalization by exploiting new statistical factors in the Multi-Aspect TF (MATF) schema. One component of the term frequency is effective for short queries, while the other performs better on long queries. The final weight is then measured by taking a weighted combination of these components, which is determined based on the length of the corresponding query. Experiments conducted on a large number of TREC news and web collections demonstrate that the proposed scheme almost always outperforms five times the state-of-the-art information retrieval models in a remarkably significant and consistent manner.

In another TF-IDF term weighting study [7], a short message service (SMS) model was developed to determine the statistical importance of a word for an SMS categorization. First, all SMSs are converted to text documents. After the pre-processing of the SMSs, a vector space model is prepared, and a weight is given to each term. This weighting method predicts the significance of a word for an SMS classification problem. Experiments reported in the article showed that this weighting method significantly improved classification accuracy.

TF–IDF was also used in sentiment analysis studies [11]. Das and his colleagues proposed text sentiment classification using TF-IDF in conjunction with Next Word Negation (NWN). They also compared the performances of a binary bag-of-words model, the TF-IDF, with TF-IDF with "next word negation" (TF-IDF-NWN) model for text classification. The results showed significant increases in accuracy compared to earlier methods.

In 2018, Qaiser et al. [33] used TF–IDF for the identification of keywords in the document and pointed out the shortcomings of this method because this method yielded unexpected results. Although the words ("go", "goes"), ("play", "playing"), ("mark", "marking") and ("year", "years") in the documents have the same word root, these words were perceived as words that altered the results. The method was also criticized for the word, coexistence, and for the control of the document in terms of the dictionary rather than meaning. It has been noted that TD–IDF can be combined with other techniques, such as Naive Bayes, to yield better results.

As stated in the previous section, the dictionary can be created in three different ways: automatic, semi-automatic, and manual. In 1993, Riloff [34] conducted a study on automatic dictionary creation. In this work, Riloff developed a system called AutoSlog, a dictionary of domain-specific concepts for extracting information from a text. When text is given to AutoSlog to be processed, AutoSlog extracts the user-requested information from this text and creates a dictionary containing the relevant words. If the text supplied to the AutoSlog system is related to the subject of the dictionary, the success of the dictionary is high. AutoSlog dictionary is a dictionary containing the domain of terrorist events created in five person-hours of effort. The AutoSlog dictionary was then compared to a handmade dictionary that was made by two talented graduate students, requiring approximately 1500 person-hours of

effort. As a result, the AutoSlog dictionary provided 98% of the performance of the handmade dictionary.

Studies on creating a language dictionary using NLP techniques attempted in different languages. Although studies referring to the English language [34, 35] are more numerous, there are also dictionary generation works in other languages, such as; Tigrinya (Semi-automatic dictionary) [1], Thai (Automatic dictionary) [39], French (Manual dictionary) [16], Arabic (Manual and Semi-automatic dictionaries) [4, 5], and Turkish (Manual dictionary) [2].

In 1999, while at Apple Computer, Silverman et al. [35] described the design and structure of the automatic Victorian dictionary created to support the speech synthesis research and development. The Victorian dictionary consists of five main chapters: (1) polyphony; (2) prosodic context; (3) repetitive speech; (4) function word sequences; and (5) continuous speech. This dictionary is designed to cover a particular aspect of each of the speech synthesis. The Victorian dictionary is written in United States (American) English. The purpose of the Victorian dictionary is to collect of semantic texts throughout the speech. The dictionary was used in the statistical estimation of the time and step models for MacinTalk 4, the next-generation text-to-speech system from Apple.

In 2005, the study by Filippo Vella et al. [37] described a technique for the automatic creation of dictionaries using the sub-symbolic representation of words in cross-language context. Semantic relationship among words of two languages was extracted from aligned bilingual text corpora. This feature was obtained by applying the Latent Semantic Analysis (LSA) technique to the matrices representing terms co-occurrences in aligned text fragments. The technique allows finding the “best translation” according to a properly defined geometric distance in an

automatically created semantic space. Experiments showed that correctness of 95% was obtained in the best case.

In 2008, the study by Bracewell et al. [9], a semi-automatic emotion dictionary was created using the English WordNet dictionary. The proposed algorithm takes as input a set of seed words manually assigned by WordNet senses and emotional information. An initial dictionary was automatically created using seed words and WordNet. Later, this initial dictionary was expanded with reference to the WordNet dictionary. Using the proposed algorithm, a dictionary of approximately 6,000 words was created from 130 seed words. Evaluation of the created dictionary was done with independently labeled corpus of emotions. In the experiments, the proposed algorithm achieved 84% accuracy. In addition, the created dictionary was also used to classify the views of news articles.

In 2010, the study by Julia Pajzs and her team [30], a semi-automatic frequency dictionary in the Hungarian language was created. The dictionary is a collection of 187.6 million words of the most commonly used Hungarian verb phrase (VPC) constructions. Dictionary words were obtained from Hungarian National Library data. Hungarian linguists verified the dictionary manually. It was stated that the obtained results were at a good level and that this dictionary will accelerate the adaptation processes of those who want to learn Hungarian.

In the specific study achieved by Kepuska and Rojanasthien [24], a data collection system was created to produce a manual speech dictionary from movies, TV series, and DVDs. The dictionary provides a lower-cost solution compared with the traditional speech dictionary acquisition method. In addition, it was stated that the collection of data and processing of the dictionary were in short time.

In 2012, an automatic dictionary creation study [39] was executed by Vorapatratorn, Suchato, and Punyabukkana. This study describes the method of automatic dictionary creation using a specific phonetic distribution. Usually, the system selects its data by downloading continuous text from the Internet through a web-based browser. The covetous algorithm is then applied to a given text to extract the appropriate words, and this process continues until the appropriate text dictionary is completed. The results of the study showed that the amount of data withdrawn from the internet achieved the target phonetic distribution and created a telephone coverage area of 99.13% approximately. This text dictionary was used to efficiently produce the speech dictionary.

In 2012, the study by Jozsef Domokos and his team [13], was presented a newly developed Romanian language pronunciation semi-automatic dictionary NaviRO. NaviRO pronunciation dictionary contained more than 100k words from the DexOnline dictionary together with their phonetic transcriptions in Speech Assessment Method Phonetic Alphabet (SAMPA), a machine-readable alphabet. It is in Festival Speech Synthesis System (F3S) dictionary format. The NaviRO dictionary was developed in two stages. In the first stage, using an artificial neural networks (ANN) system having a parallel structure of 30 neural networks developed a 5k word pronunciation dictionary. The automated grapheme-to-phoneme transcription system was tested on a small 1k word, hand-built database. The trained system was able to perform grapheme-to-phoneme transcription with an accuracy of 92.83%, calculated at the phoneme level. In the second stage, built the 100k+ words NaviRO dictionary using Dictionary Maker, a software application created to facilitate the creation of an electronic pronunciation dictionary in a target language starting from the previously created 5k dictionary and a 100k wordlist. The more than 100.000 Romanian words were

collected from the DexOnline dictionary, the largest online dictionary for the Romanian language.

In 2013, the study by Herrero-Zazo et al. [17], an annotated automatic dictionary containing Pharmacological substances and Drug-drug interactions (DDI) was created to reduce the time healthcare professionals spending for review of the biomedical literature and to supply an annotated dictionary for pharmacological substances. The created dictionary had a manually annotated structure consisting of 792 texts selected from the DrugBank database and 233 other Medline abstracts. This fined-grained dictionary was annotated with a total of 18,502 pharmacological substances and 5028 DDIs, including both pharmacokinetics (PK) as well as pharmacokinetics (PD) interactions. In tests, the task of recognizing and classifying pharmacological names achieved an F1 score of 71.5%, while the detection and classification of DDIs achieved an F1 score of 65.1%. These results showed that the created dictionary had sufficient quality in the field of Pharmacovigilance.

In 2013, the study by Behmadi Moghaddas et al. [8], was created a manual dictionary to overcome the obstacles to performing automatic text summarization in Persian. The reason for this obstacle is that there is no standard dictionary for evaluating Persian text summaries. To solve this problem, a dictionary named Pasokh was created. The Pasokh dictionary is a dataset of numerous Persian news documents on various topics. 2000 person-hours were spent to produce this dataset. Then, the success of summarizing was observed by giving single and multiple documents to the Pasokh dictionary. Pasokh dictionary achieved a success rate of 83.56% in single-document summaries and 60.24% in multi-document summaries.

In 2017, Aktas and his colleagues [2], a detailed WordNet literature research conducted. A manual dictionary containing Turkish informatics terms was created. The words in the dictionary are grouped according to their relationship, similarity, and class. It

was stated that the resulting dictionary was larger and more comprehensive than other information dictionaries created up to date.

One of the latest works was completed by Vijay and his team [38]. They created a Hindi-English code-mixed semi-automatic dictionary using tweets published online for the last eight years. To create this dictionary, tweets were taken from Twitter, using the Twitter Python API, which primarily uses the advanced search option in Twitter. All information, such as received tweets, timestamps, Uniform Resource Locator (URL), text, user, retweets, answers, full name, identities, and likes were converted to the JavaScript Object Notation (JSON) format. A comprehensive semi-automated procedure was performed to remove all noisy tweets. After completion of these steps, a dictionary of 2866 words was created, and dictionary words were classified as happiness, sadness, anger, surprise, hate, and multiple emotions. Emotional analysis was performed for the tweets published online, and a 58.2% accuracy was obtained.

In 2018, the study by Fatima-Al-Raisi and her team [3], was presented a new monolingual automatic dictionary for Arabic generated from translating a parallel dictionary of English-French using the Google Translate API. There are no identical words in any of the entries (sentence pairs) in the dictionary. The dictionary contains 100K word pairs. A sample was evaluated by two native speakers of Arabic and was found to be grammatical or nearly grammatical and semantically coherent. Corresponding sentences were found to be both similar in meaning and varied in surface form.

In 2019, a Hinglish (Hindi-English) code change database semi-automatic dictionary called IITG-HingCoS dictionary was developed by Sreeram et al. [14] at Guwahati Institute of Technology, India (IITG). Code-switching was a phenomenon in linguistics, specifically referring to the use of two or more languages

within the same discourse. The created dictionary consisted of code replacement text data a total of 0.58 million words and 25.988 sentences. In addition, the dictionary also contained 25 hours of matched speech data, corresponding to 9251 code replacement phrases, covering a vocabulary of 6542 words. The dictionary was stated to be quite useful for language modeling and automatic speech recognition (ASR) tasks.

In 2021, Alla et al. [6] presented experimentally the methods used, and the difficulties encountered in cohort selection for establishing a clinical patient dictionary in Electronic Health Record (EHR) systems. In the study, it was expressed that basic patient information was usually kept in the form of unstructured clinical notes and it was difficult to process. To overcome this problem, it was aimed to create an automatic patient dictionary by using cancer pathology reports. Using the HL7 (Health Level-7) message standard, 2100 XML-type pathology reports from 1833 (518 males and 1313 females) cancer patients were taken as reference in this study. These documents were separated using NLP techniques, and the dictionary was created by taking the relevant parts from the XML data and adding them to the clinical patient dictionary, according to the prepared manual patterns. It was declared that although the results obtained are generally at an acceptable level, the clinical coding and metadata standards that vary between different hospitals and mostly the low EHR data quality prevents the development of a high-quality dictionary.

In 2022, the study by Aikaterini Kant et al. [23], a methodology was presented for creating custom dictionary word lists for dictionary-based password cracking attacks, starting with a single context seed word. In the study, it was stated that context is very important for a user during password selection and there are not yet automated approaches that can extract and use contextual information during password cracking processes. In addition, it was

stated that people often choose passwords that are related to the subject of the website/system or that are thematically close to that subject when setting passwords. Therefore, using a custom dictionary will provide a significant advantage in password cracking. In the proposed methodology, custom dictionaries were created for any topic of interest that may be required, leveraging the power of structured information found in Wikipedia and DBPedia. Dictionaries are fully customizable. In this way, it was aimed to perform context-based password-cracking attacks. Experiments showed that using the proposed approach with existing approaches resulted in 15.5% more passwords being cracked than existing approaches.

## 3. Method

In order to identify the meaningful words of the documents given to the system, the documents are pre-processed initially. After meaningful terms are determined, they are now candidates for the dictionary and web search. The general system schema is given in Fig. 1.

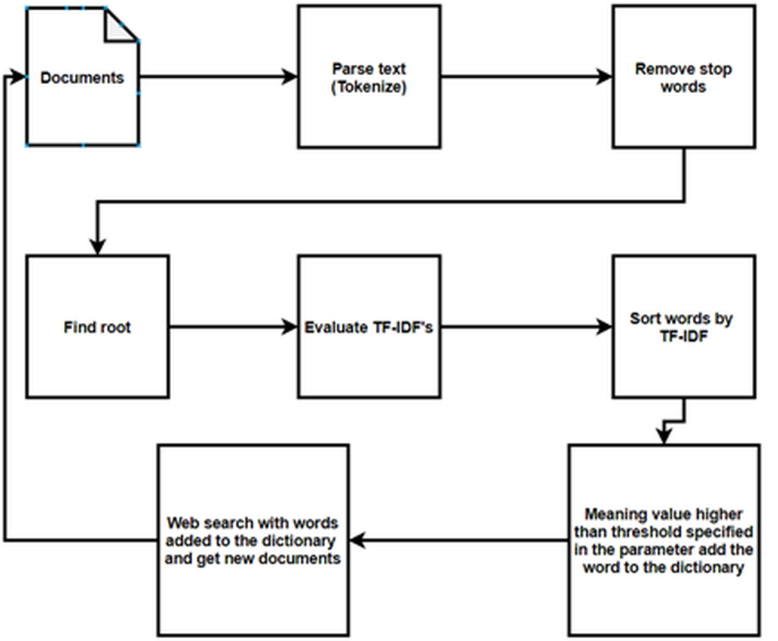



***Figure 1.*** *General structure of automatic dictionary creation*

### 3.1. Data pre-processing

Data pre-processing is one of the most important preliminary steps of NLP and is often troublesome but affects the success of the algorithm. In pre-processing, it is necessary to remove prepositions, conjunctions, exclamations, letters, and words with no categorical meaning. These words, called stop words, must be filtered in the first step of data pre-processing. Data pre-processing is applied to each document, and it is used for noise removal, lexicon normalization, and object standardization.

After this process, the document words are converted to the lower-case format. It is then necessary to separate the words from the suffixes and find their root forms. The main purpose of stemming is to reduce different grammatical forms/word forms of a selected word, such as its noun, adjective, verb, or adverb, to its root form. The most common rule-based stemming algorithm for the English

language was developed by Porter, and it is still the most commonly used method to date. The most commonly known and easy-to-use Porter Stemmer algorithm [32] was used in this project. Examples successfully applied to stemming are given by Jivani [22]. Fig. 2 lists the stemming algorithms in NLP.

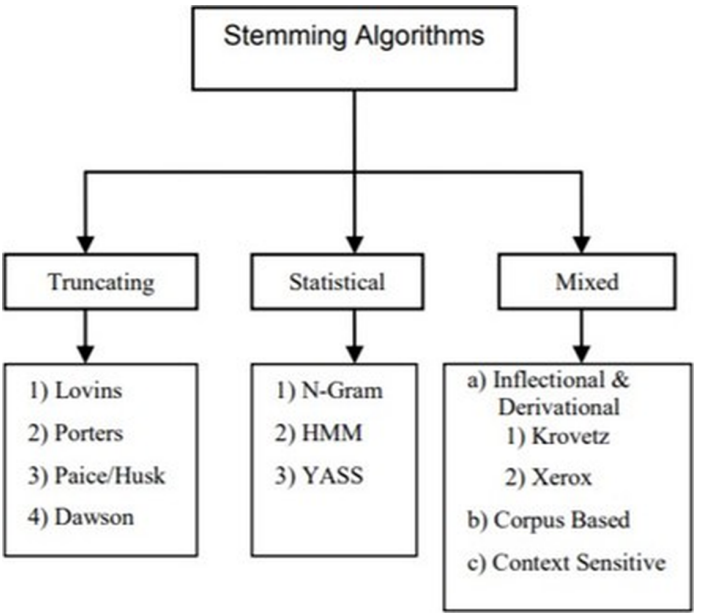



***Figure 2.*** *Types of stemming algorithms*

### 3.2. Selection of keywords

At this stage, the main system task is to find meaningful words from both the initial given documents and the documents returned from the web search. In this study, the TF–IDF was applied to find the meaning values of words. TF–IDF consists of combination of two different concepts: (1) term frequency and (2) inverse document frequency. The TF–IDF is commonly used to weight each word in the text document according to its level of uniqueness. In other words, the TF–IDF captures the relationship between words, text documents, and specific categories [12].

Suppose we have a document, “t1”, containing 5000 words, and the word “alpha” is present in the document exactly 10 times. It is a very well-known fact that the total length of documents can vary from very small to large, so it is highly possible that a term may occur more frequently in large documents in comparison to small documents. Thus, in order to normalize document length, the occurrence of any term in a document is divided by the total terms present in that document to yield the term’s frequency. So in this case, the term frequency of the word “alpha” in the document “t1” is calculated according to Equation (1).

$$\mathrm{TF}(\mathrm{alpha}) = \frac{10}{5000} = 0.002 \tag{1}$$

Inverse document frequency (IDF) assigns less weight to common words and more weight to less common words. For example, if we have 10 documents and the term “technology”, is available in five of these documents, the IDF value is calculated according to Equation (2).

$$\mathrm{IDF}(\mathrm{technology}) = \log_e \frac{10}{5} = 0.3010 \tag{2}$$

The TF–IDF calculation is equal to the product of the TF and IDF values. TF (w) and IDF (w) values given in Equation (3).

$$\text{TF-IDF}(w) = \mathrm{TF}(w) \times \mathrm{IDF}(w) = 0.002 \times 0.3010 = 0.000602 \tag{3}$$

First, all word frequencies in the document set are recorded in the database. Using the formulas mentioned above, TF, IDF, and TF–IDF values of each word are then calculated. An example of the metric values for the given document set is shown in Table 1. After calculating these values, the words with a meaning value above the threshold are added to the dictionary. The resulting dictionary hash similarity values obtained by adding different threshold meaning values are presented in detail in section 4.

***Table 1.*** *Financial words and TF-IDF scores after the first iteration*

| Document Id | Word | TF | IDF | TF-IDF |
|---|---|---|---|---|
| 1 | season | 0.0422 | 1 | 0.0422 |
| 1 | yard | 0.0578 | 0.6931 | 0.0400 |
| 1 | pass | 0.0533 | 0.6931 | 0.0370 |
| 1 | player | 0.0222 | 1 | 0.0222 |
| 1 | game | 0.0200 | 0.6931 | 0.0139 |
| 1 | touchdown | 0.0200 | 0.6931 | 0.0139 |
| 1 | touchdown | 0.0200 | 0.6931 | 0.0139 |

### 3.3. Creating the first dictionary words from the starting document

When the document related to the subject of the desired dictionary is initially given to the system, the meaningful words of these seed document are determined using TF–IDF. Only the highest TF–IDF value is added to the dictionary. The web search is then started with this word. However, the rule of determining meaningful candidate words for insertion into the dictionary through the web search is not only the most meaningful word but also the words in which the TF–IDF values are above 0.03.

### 3.4. Determination of similarity value

When candidate meaningful words are selected after a web search, a similarity value using WordNet similarity is then calculated for each word with respect to the dictionary. Finally, the words with similarities above the similarity threshold value are added to the dictionary. For example, after finding meaningful words for web search results, suppose that three words with the highest TF–IDF values will be added to the dictionary. Suppose these three words are “student", "math", and "lesson". Let the words in the dictionary be “team” and “ball”. A loop is then used to calculate the WordNet similarity of each meaningful word with the words in the dictionary. If we take the word “football”:

WordNet Similarity (“football”, “team”) = 0.35 WordNet Similarity (“football”, “ball”) = 0.30 Total WordNet Similarity (“football”) = 0.35 + 0.30 = 0.65 is the result.

The average WordNet similarity (AWS) value of the word is then calculated using Equation 4.

$$\mathrm{AWS}(w) = \frac{\mathrm{TotalWordNetSimilarity}}{\mathrm{DictionaryWordCount}} = \frac{0.65}{2} = 0.325 \qquad (4)$$

If the average WordNet similarity parameter value is defined as 0.35 in the application, then the word is not added to the dictionary; on the other hand, if the similarity parameter value is 0.30, the word is added to the dictionary.

WordNet is the English dictionary database in the Cognitive Science Laboratory of Princeton University. Names, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each of which expresses a separate concept. The hypernym feature, a more abstract and general meaning of a word, is used in this database. For example, by looking for the word hypernym for the word cat in the WordNet database, the concept of this word can be determined as the animal. WordNet is free and can be downloaded publicly. The structure of WordNet is therefore a useful tool for computational linguistics and NLP. Fig. 3 shows the WordNet hierarchy [40].

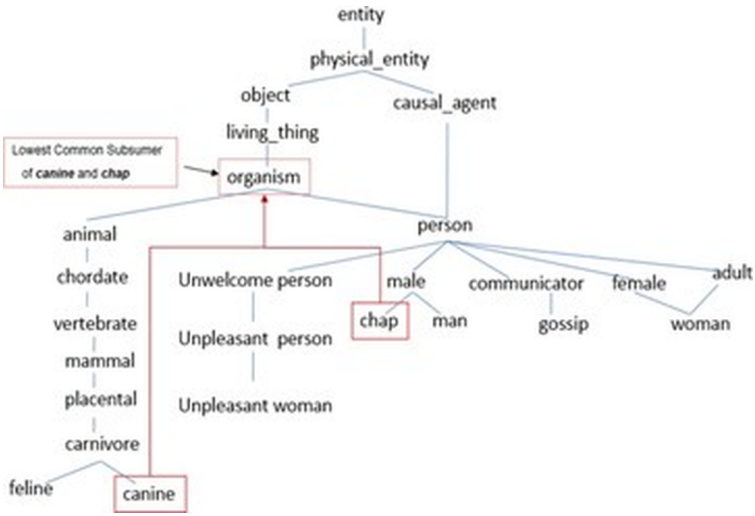



***Figure 3.*** *Sample of WordNet hierarchy*

### 3.5. Web search process

The most meaningful word inserted into the dictionary for the last iteration is used in the next web search together with the synonym words obtained from WordNet. For example, if the word added to the dictionary is “student”, the synonym for “student” is “pupil”. In the web search, “student” and “pupil” are used together.

The Azure Cognitive Web Search Interface is used to search on the web. The words to be used on the web search are combined with the space character and given as a parameter to the Azure Web Search Service. For example, the words “Ronaldinho”, “Barcelona”, and “El Classico” are listed consecutively, and these words are used together as “Ronaldinho Barcelona El Classico” in the web search. The web search with these words will result in more than one document. Among these documents, a single document that has not been previously processed is selected. Pre-processing steps are then applied to this document, and new meaningful words are obtained. Thus, the dictionary is constantly expanding with new documents, and the dictionary can be prevented from repeating itself. This

process continues until the dictionary reaches the number of dictionary words in which a parameter value is initially given by the user to the system. Thus, a continuous cycle is provided in the system.

### 3.6. Determination of general similarity value of dictionary

In this step, the similarity of the dictionary is determined using the SimHash algorithm. The SimHash algorithm is an algorithm used to find similarities between files or websites in applications, such as a search engines, especially in text processing. The SimHash algorithm views two files as vectors and tries to find the cosine link between these vectors.

SimHash is a hash function, and the more similar its property is to the text input, the smaller the Hamming distance of the hash values (Hamming distance is the number of locations in which the corresponding symbols are different). The algorithm works by dividing the text into pieces and combining each piece with a selected function. Each mixed set is represented as a binary vector, and the bit value is converted to +1 or –1 depending on whether the bit value is 1 or 0. In order to obtain SimHash, all bit vectors as bit-wise are collected. Finally, if the total is negative, the bits that result as 1 are otherwise set to 0.

The SimHash algorithm has been used in many different studies for similarity detection. For example, in studies by Jiang and Pi [21, 31], the SimHash algorithm is used to obtain document similarity. Fig. 4 illustrates the working procedure of the SimHash algorithm [18].

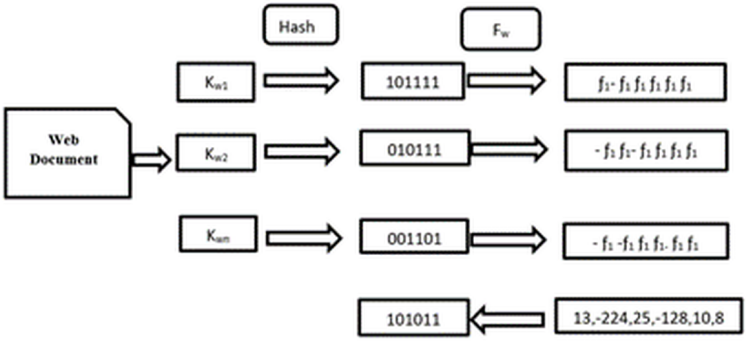



***Figure 4.*** *Working procedure of the SimHash algorithm*

## 4. Experimental Results

For all dictionaries, documents belonging to the sports data and containing HTML content used in the study [36] were given to the system as a starting point. First, the number of dictionary words was set at 25, 50, and 100. The effect of the meaning value of a word calculated by the TF–IDF was then evaluated for candidate word selection for the dictionary. Finally, the effect of the number of documents given initially was used in the experiment. For each experiment, the dictionary similarity values were calculated and discussed.

### 4.1. Experiment I

In Table 2, the hash similarity value for 25 words limited dictionary initialized by only 1 sport (badminton) document is given.

***Table 2.*** *Hash similarity value for the 25 words limited dictionary initialized by only one sports document*

| Parameter Name | Parameter Value |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.03 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | 75.2 |

The similarity value of the 25 words dictionary was shown to be higher than the dictionaries with 50 and 100 words. The most important reason for achieving this similarity value was that the new documents obtained from the web causes deviation from the original reference document topic. Table 3 lists the words obtained for parameters given in Table 2.

***Table 3.*** *Twenty-five words limited dictionary initialized with only one sports document*

| Sports Words | | | | |
|---|---|---|---|---|
| academy | debut | game | people | set |
| badminton | entry | injury | player | shoe |
| bwf | event | medal | racket | shop |
| committee | exercise | meter | sear | sport |
| court | federation | Olympic | select | world |

In Table 4, the hash similarity value for the 50 words limited dictionary initialized by only 1 sport (badminton) document is given.

***Table 4.*** *Hash similarity value for the 50 words limited dictionary initialized by only one sports document*

| Parameter Name | Parameter Value |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.03 |
| Dictionary Maximum Number of Words | 50 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | 73.8 |

The similarity value of the 50 words dictionary was higher than the dictionary with 100 words. Obviously, when all parameters were kept constant, and only the number of dictionary words was increased, the dictionary similarity value decreased. Table 5 lists the words obtained for the parameters given in Table 4.

***Table 5.*** *Fifty words limited dictionary initialized with only one sports document*

| **Sports Words** | | | | |
|---|---|---|---|---|
| academy | country | Excel | player | shoe |
| arcadia | court | Feder | policy | shop |
| badminton | cutter | game | postage | sport |
| Blais | dark | injury | racket | technique |
| box | debut | league | school | tennis |
| brand | dimension | ligament | sear | weight |
| bwf | enjoy | medal | select | wayn |
| committee | entry | member | service | world |
| company | equip | meter | set | wrist |
| cookie | event | Olympic | ship | Yonex |

In Table 6, the hash similarity value for the 100 words limited dictionary initialized by only one sport (badminton) document is given.

***Table 6.*** *Hash similarity value for the 100 words limited dictionary initialized by only one sports document*

| **Parameter Name** | **Parameter Value** |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.03 |
| Dictionary Maximum Number of Words | 100 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | 68.7 |

The similarity value of the 100 words dictionary was lower than the similarity values of the 25 and 50 words limited dictionaries. Using this result, we could make an inference: When the maximum

number of words in the dictionary increases, deviations occur in the dictionary after a certain number of words. The factor that causes the deviation originates from the web search that was done with this word after adding an unmeaningful word to the dictionary. In order to prevent deviations in the dictionary, the dictionary should be checked periodically, and words relating to this document should be added to the dictionary based on the initial reference document. Table 7 lists the words obtained for the parameters given in Table 6.

***Table 7.*** *One-hundred words limited dictionary initialized with only one sports document*

| **Sports Words** | | | | |
|---|---|---|---|---|
| academi | company | group | och | shuttlecock |
| address | cookie | guest | Olympia | sign |
| Alberta | country | har | oppon | Singapore |
| answer | court | hotel | pain | skill |
| associ | cutter | image | partner | sport |
| backhand | dan | Inform | player | team |
| badminton | debut | ingen | point | technique |
| ball | double | injury | policy | thesen |
| basic | draw | job | procedure | tournament |
| bird | entry | knee | program | use |
| box | equip | language | question | vatn |
| brand | event | league | racket | video |
| build | excel | ligament | read | way |
| bwf | experiment | line | sear | weight |
| Canada | fact | match | season | wib |
| card | fast-paced | medal | select | world |
| championship | fat | member | service | wrist |
| club | federation | merge | set | year |
| coach | format | meter | shoe | Yonex |
| committee | game | ntrue | shop | zone |

### 4.2. Experiment II

In this experiment, the effect of the meaning threshold value of a word was examined. If the meaning threshold value was selected to be 0.03, words with meaning values greater than this meaning threshold value became candidates for the dictionary. If this threshold was applied through the selection of dictionary words, higher dictionary similarity values were obtained. On the other hand, when words with a meaning value less than 0.03 are added to the dictionary, lower dictionary similarity values are obtained. In the following experiments, the threshold value for the TF–IDF was selected as 0.02, and 25 and 50 words limited dictionaries were created in order.

In Table 8, the hash similarity value for the 25 words limited dictionary initialized with only one sports (badminton) document is given.

***Table 8.*** *Hash similarity value for the 25 words limited dictionary initialized by only one sports document*

| Parameter Name | Parameter Value |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.02 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | 62.5 |

One sports document (badminton) was again given to the system, and as a result of the calculation with the TF–IDF, a 25 words-limited dictionary was produced using the words with a meaning value equal to or lower than 0.02. The overall similarity value of the final dictionary was calculated using the hash similarity method, and a 62.5% similarity was obtained. This similarity value was lower than the similarity value of all of the dictionaries obtained by adding the words with the meaning value over 0.03. It was noted that the addition of words lower than the meaningful threshold value

to the dictionary caused deviations and decreased the similarity of dictionary words. Table 9 includes dictionary words obtained in this experiment.

***Table 9.*** *Twenty-five words limited dictionary initialized with only one sports document*

| **Sports Words** | | | | |
|---|---|---|---|---|
| badminton | committee | Feder | Olympic | sport |
| ball | court | game | player | state |
| bwf | day | language | point | street |
| city | dimens | meter | quarter-final | tennis |
| coach | exercise | nn | sign | world |

In Table 10, the hash similarity value for the 50 words limited dictionary initialized by only one sports (badminton) document is given.

***Table 10.*** *Hash similarity value for the 50 words limited dictionary initialized with only one sports document*

| **Parameter Name** | **Parameter Value** |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.02 |
| Dictionary Maximum Number of Words | 50 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | 59.2 |

The overall similarity of this dictionary was calculated using the Hash similarity method and was equal to 59.2%. This similarity value is lower than the expected similarity value of the 25 words limited dictionary described in Experiment I. Table 11 includes the dictionary words obtained in this experiment.

***Table 11.*** *Fifty words limited dictionary initialized with only one sports document*

| **Sports Words** | | | | |
|---|---|---|---|---|
| academy | committee | event | Olympic | shoe |
| article | competition | Feder | play | side |
| assist | cookie | feedback | player | Singapore |
| associate | court | game | point | sport |
| badminton | cup | group | policy | team |
| book | cutter | Klee | racket | tennis |
| box | descript | list | read | tournament |
| bwf | draw | medal | select | world |
| champion | editor | member | service | year |
| championship | entry | meter | set | chance |

## 4.3. Experiment III

In this experiment, the effect of the number of documents fed to the system initially was studied. In the following experiments, two different dictionaries of 25 words were created by giving two and three initial documents to the system. In Table 12, the hash similarity value for the 25 words limited dictionary initialized by two sports (badminton)- related documents are given.

***Table 12.*** *Hash similarity value for the 25 words limited dictionary initialized with two sports documents*

| **Parameter Name** | **Parameter Value** |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.03 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 2 |
| Dictionary Similarity Value % | 62.7 |

When all parameters were kept constant, however, the number of documents initially fed to the system were changed, and the similarity value of the dictionary for a small number of initial documents then becomes higher. This finding can be explained by

deviations in the dictionary words based on the initial given documents. Table 13 lists the words obtained for the parameters given in Table 12.

***Table 13.*** *Twenty-five words limited dictionary initialized with two sports documents*

| **Sports Words** | | | | |
|---|---|---|---|---|
| announce | court | canada | person | team |
| associate | entry | dimense | rule | ticket |
| badminton | event | double | service | video |
| championship | game | inform | side | yonex |
| contact | bwf | line | sport | zone |

In Table 14, the hash similarity value for the 25 words limited dictionary initialized by three sports (badminton) documents is given.

***Table 14.*** *Hash similarity value for the 25 words limited dictionary initialized by three sports documents*

| **Parameter Name** | **Parameter Value** |
|---|---|
| Dictionary Similarity Threshold of the Word | 0.03 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 3 |
| Dictionary Similarity Value % | 60.3 |

The similarity value of the 25 words-limited dictionary created with three initial documents was the lowest dictionary similarity value, including almost the dictionaries shown in experiments 1 and 2. It should be noted that even if the TF–IDF meaning value is set to 0.03, dictionaries with a high success rate cannot be obtained when too many initial documents are given. Table 15 lists the words obtained for the parameters given in Table 14.

***Table 15.*** *Twenty-five words limited dictionary initialized with three sports documents*

| **Sports Words** | | | | |
|---|---|---|---|---|
| associate | badminton | entry | partner | service |
| august | championship | event | side | time |
| contact | dog | game | player | world |
| court | change | hour | point | year |
| date | dont | inform | score | weight |

## 5. Conclusion

In this study, meaningful words of the seed documents were calculated using the TF–IDF, and a specific domain dictionary was automatically created. Through this study, the effect of establishing a similarity threshold between a candidate word to be inserted into the dictionary, dictionary size, and the effect of the initial number of documents were researched. As a result, we can conclude three findings from these experiments:

- The first one indicates that when the dictionary size grows, the similarity of the dictionary then decreases intuitively as expected. This can be explained by less deviation in the dictionary because the number of dictionary words was small.
- The second one is to establish a bigger threshold for the meaningful value of a word to add to the dictionary to enhance the dictionary similarity. This is a result of selecting more meaningful (similar) words.
- The third one indicates that when the number of initial documents given increases, dictionary similarity significantly decreases as a result of the differences of the contents of the given documents.

The work was compared with similar studies, and it is obvious that the results of the study are quite successful in terms of

average results, dictionary creation, and growth rate. The study [38] mentioned in the literature review section achieved successful results in automatic dictionary creation in two studies. However, unlike [38], the web search section was included in this work. This work is more valuable than the study of [38] in terms of working on instant and current data. Moreover, the WordNet similarity calculation is used in this work to determine whether the words to be added to the dictionary are related to the subject to be created. In general, the following conclusions can be drawn from the results of the work.

- The set of documents should be related to the dictionary you want to create. If the system is presented with a set of documents that are not associated with the desired dictionary, the resulting dictionary will contain meaningless words.
- Once meaningful words are found, the words above the dictionary similarity threshold value are added to the dictionary. When the similarity threshold value is increased, the speed of dictionary creation decreases, while the Hash similarity of the final dictionary increases.
- Web search is made with meaningful words added to the dictionary. When searching the web with a certain percentage of these words, not all meaningful words, the success rate will change positively.
- Querying with the last words added to the dictionary for processing different data continuously during the web search process will prevent duplication of the dictionary. It is therefore useful to mark the last words in the dictionary.

The following studies, which are thought to contribute to the literature, will be discussed.

- Since the suggested system operates in a continuous growth manner, the number of iterations in which the system approaches saturation can be determined.
- A WordNet similarity method was used to calculate the similarity value between the words in the dictionary. According to the WordNet similarity result, it is decided whether or not to add the word to the dictionary. Different similarity techniques could also be examined. For example, the Word2Vec similarity method and WordNet similarity method could be compared. Structural differences between the two dictionaries could then be revealed.
- Once it has been determined to add a word into the dictionary, all synonyms of this word could be added to the dictionary together. The SimHash algorithm could be calculated by comparing the results.

## References

[1] Abera, H., H/Mariam, S., 2018. Design of a Tigrinya language speech corpus for speech recognition, in: Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing, Association for Computational Linguistics, Santa Fe, New Mexico, USA. pp. 78–82. URL: https://aclanthology.org/W18-3811.

[2] Aktaş, Y., Yılmaz Ince, E., Çakir, A., 2017. Doğal dil İşleme kullanarak bilgisayar ağ terimlerinin wordnet ontolojisinde uyarlanması .

[3] Al-Raisi, F., Lin, W., Bourai, A., 2018. A monolingual parallel corpus of arabic. Procedia Computer Science 142, 334–338. doi:10.1016/j.procs.2018.10.487.

[4] Alansary, S., Nagi, M., 2014. The international corpus of arabic: Compilation, analysis and evaluation, pp. 8–17. doi:10.3115/v1/W14-3602.

[5] Alansary, S., Nagi, M.H., Adly, N., 2007. Building an international corpus of arabic ( ica ) : Progress of compilation stage.

[6] ALLA, N.L.V., CHEN, A., BATONGBACAL, S., NEKKANTTI, C., Dai, H.J., JONNAGADDALA, J., 2021. Cohort selection for construction of a clinical natural language processing corpus. Computer Methods and Programs in Biomedicine Update 1, 100024. URL: https://www.sciencedirect.com/science/article/pii/S2666990021000239, doi:https://doi.org/10.1016/j.cmpbup.2021.100024.

[7] Altalib, G., Hassan, H., 2013. A study on analysis of sms classification using tf-idf weighting. International Journal of Computer Networks and Communications Security 1, 189–194.

[8] Behmadi Moghaddas, B., Kahani, M., Toosi, S.A., Pourmasoumi, A., Estiri, A., 2013. Pasokh: A standard corpus for the evaluation of persian text summarizers, in: ICCKE 2013, pp. 471–475. doi:10.1109/ICCKE.2013.6682873.

[9] Bracewell, D., 2008. Semi-automatic creation of an emotion dictionary using wordnet and its evaluation. 2008 IEEE Conference on Cybernetics and Intelligent Systems , 1385–1389.

[10] Chabi, A.H., Kboubi, F., Ahmed, M.B., 2011. Thematic analysis and visualization of textual corpus URL: https://arxiv.org/abs/1112.2071, doi:10.48550/ARXIV.1112.2071.

[11] Das, B., Chakraborty, S., 2018. An improved text sentiment classification model using tf-idf and next word negation. ArXiv abs/1806.06407.

[12] Deepshikha Patel, M.B., 2013. Mobile sms classification, in: International Journal of Soft Computing and Engineering (IJSCE). doi:10.1007/978-3-642-38824-8_25.

[13] Domokos, J., Buza, O., Toderean, G., 2012. 100k+ words, machine-readable, pronunciation dictionary for the romanian language, in: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 320–324.

[14] Ganji, S., Dhawan, K., Sinha, R., 2019. Iitg-hingcos corpus: A hinglish code-switching database for automatic speech recognition. Speech Communication 110, 76–89. URL: https://www.sciencedirect.com/science/article/pii/S0167639318304217, doi:https://doi.org/10.1016/j.specom.2019.04.007.

[15] Gholamrezazadeh, S., Salehi, M.A., Gholamzadeh, B., 2009. A comprehensive survey on text summarization systems. 2009 2nd International Conference on Computer Science and its Applications , 1–6.

[16] Grabar, N., Claveau, V., Dalloux, C., 2018. Cas: French corpus with clinical cases, in: Louhi@EMNLP.

[17] Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., Declerck, T., 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. Journal of Biomedical Informatics 46, 914–920. URL: https://www.sciencedirect.com/science/article/pii/S1532046413001123, doi:https://doi.org/10.1016/j.jbi.2013.07.011.

[18] Ho, P.T., Kim, S.R., 2014. Fingerprint-based near-duplicate document detection with applications to sns spam detection. International Journal of Distributed Sensor Networks 10, 612970. URL: https://doi.org/10.1155/2014/612970, doi:10.1155/2014/612970,arXiv:https://doi.org/10.1155/2014/612970.

[19] Hovy, E., Lin, C.Y., 1998. Automated text summarization and the summarist system, Association for Computational Linguistics, USA. p. 197–214. URL: https://doi.org/10.3115/1119089.1119121, doi:10.3115/1119089.1119121.

[20] Indyk, P., Motwani, R., 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, NY, USA. p. 604–613. URL: https://doi.org/10.1145/276698.276876, doi:10.1145/276698.276876.

[21] Jiang, Q., Sun, M., 2011. Semi-supervised simhash for efficient document similarity search, in: Annual Meeting of the Association for Computational Linguistics.

[22] Jivani, A., 2011. A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl. 2, 1930–1938.

[23] Kanta, A., Coisel, I., Scanlon, M., 2022. A novel dictionary generation methodology for contextual-based password cracking. IEEE Access 10, 59178–59188. doi:10.1109/ACCESS.2022.3179701.

[24] Kepuska, V., Rojanasthien, P., 2011. Speech corpus generation from dvds of movies and tv series. Journal of International Technology and Information Management .

[25] Koeva, Stoyanova, T., Leseva, 2016. Ssemi-automatic compilation of the dictionary of bulgarian multiword expressions, in: Proceedings of GLOBALEX 2016: Lexicographic Resources for Human Language Technology.

[26] Lee, K.M., 2012. Locality-sensitive hashing techniques for nearest neighbor search. International Journal of Fuzzy Logic and Intelligent Systems 12, 300–307. doi:10.5391/IJFIS.2012.12.4.300.

[27] Maksutov, A.A., Zamyatovskiy, V.I., Vyunnikov, V.N., Kutuzov, A.V., 2020. Knowledge base collecting using natural language processing algorithms, in: 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 405–407. doi:10.1109/EIConRus49466.2020.9039303.

[28] Nguyen, T., Shirai, K., 2013. Text classification of technical papers based on text segmentation. doi:10.1007/978-3-642-38824-8_25.

[29] Paik, J.H., 2013. A novel tf-idf weighting scheme for effective ranking, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA. p. 343–352. URL: https://doi.org/10.1145/2484028.2484070, doi:10.1145/2484028.2484070.

[30] Pajzs, J., Sass, B., 2010. Towards semi-automatic dictionary making – creating the frequency dictionary of hungarian verb phrase constructions.

[31] Pi, B., Fu, S., Wang, W., Han, S., 2009. Simhash-based effective and efficient detecting of near-duplicate short messages.

[32] Porter, M.F., 2001. Snowball: A language for stemming algorithms.

[33] Qaiser, S., Ali, R., 2018. Text mining: Use of tf-idf to examine the relevance of words to documents. International Journal of Computer Applications .

[34] Riloff, E., 1993. Automatically constructing a dictionary for information extraction tasks, in: Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press. p. 811–816.

[35] Silverman, Anderson, Bellegarda, Lenzo, Naik, 1999. Design and collection of a corpus of polyphones and prosodic contexts for speech synthesis research and development.

[36] Turan, M., Ögtelik, S., 2018. İngilizce dokümanlarda tema ve alt kavramlar tespit modeli. Düzce Üniversitesi Bilim ve Teknoloji Dergisi , 754 – 764doi:10.29130/dubited.420104.

[37] Vella, F., Pilato, G., Motisi, I., Gaglio, S., 2005. Automatic dictionary creation by sub-symbolic encoding of words, pp. 113–119. doi:10.1007/11731177_17.

[38] Vijay, D., Bohra, A., Singh, V., Akhtar, S.S., Shrivastava, M., 2018. Corpus creation and emotion prediction for Hindi-English code-mixed social media text, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, New Orleans, Louisiana, USA. pp. 128–135. URL: https://aclanthology.org/N18-4018, doi:10.18653/v1/N18-4018.

[39] Vorapatratorn, S., Suchato, A., Punyabukkana, P., 2012. Automatic online text selection for constructing text corpus with custom phonetic distribution. 2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE) , 6–11.

[40] Wei, Chang, 2015. Measuring word semantic relatedness using wordnet-based approach. Journal of Computers , 252–259.

# CHAPTER 3

# ARTIFICIAL INTELLIGENCE ENHANCED CYBERSECURITY IN DIGITAL EDUCATION

**ÖMER SEVİNÇ**[1]

**Abstract** Digital education depends on learning management systems, cloud collaboration, remote assessment, student information systems, and generative artificial intelligence services. This dependence expands the attack surface and exposes institutions to credential theft, ransomware, data leakage, academic integrity attacks, and manipulation of AI models. This chapter examines how machine learning, behavioral analytics, natural language processing, and automated response can strengthen prevention, detection, and recovery in educational environments. It also argues that AI-based controls must remain proportionate to learners' rights because continuous monitoring can create privacy, bias, and due-process risks. A layered framework is proposed that combines identity security, data governance, AI model assurance, human review, and cyber-literacy.

**Keywords** artificial intelligence, cybersecurity, education, learning management systems, student privacy, generative AI

1 Dr., Ondokuz Mayıs University, Vezirköprü Vocational School, Department of Computer Programming.

## Introduction

Universities and schools now operate as data-intensive digital organizations. A single learner may use an institutional identity to enter a learning management system, a video platform, a cloud drive, a library database, an examination service, and an AI tutor. The resulting ecosystem improves access but connects systems with different owners, security controls, and retention rules. Attackers exploit that fragmentation through phishing, password reuse, malicious browser extensions, compromised third-party applications, and ransomware. Educational institutions are attractive targets because they hold identity records, financial information, research data, and information about minors while maintaining open networks and large, frequently changing user populations.

Artificial intelligence has two roles in this environment. Defenders can use AI to identify anomalous access, prioritize vulnerabilities, classify malicious messages, and accelerate incident triage. Adversaries can use the same technology to produce convincing phishing messages, imitate instructors, automate reconnaissance, or generate malicious code. The policy question is therefore not whether education should use AI for cybersecurity, but where automation improves security without imposing disproportionate surveillance or transferring educational decisions to opaque systems (UNESCO, 2023; ENISA, 2025).

## Threat Model for Digital Learning

The first risk domain is identity. Compromised accounts can expose coursework, grades, financial details, and research resources. Conventional rules may flag every foreign login, but behavioral models can combine device history, time, location, session velocity, and access sequence to estimate risk. A high-risk session can trigger step-up authentication rather than an immediate lockout, reducing disruption for legitimate travelers.

The second domain is data and application security. Learning platforms receive file uploads, rich-text input, plug-ins, and links to external services. These functions create opportunities for malware delivery, insecure integrations, and accidental disclosure. Generative AI introduces additional channels: students or staff may paste personal or confidential material into public tools, and retrieval-augmented systems may reveal restricted documents when access filters are weak. Prompt injection can cause an AI assistant to disregard its intended instructions or expose connected data.

The third domain concerns integrity and availability. Ransomware can interrupt teaching, admissions, payroll, and assessment. Deepfakes and synthetic messages can impersonate administrators during emergencies. Examination systems may face denial-of-service attacks, question-bank theft, or manipulation of proctoring data. These events affect more than information technology; they can alter grades, delay graduation, or deny access to education.

**AI Enabled Defensive Capabilities**

AI-supported email security can analyze language, sender relationships, URLs, and attachment behavior to identify targeted phishing that signature-based filters miss. Endpoint and network models can establish baselines for laboratories, classrooms, and administrative devices, then identify unusual encryption activity, lateral movement, or large transfers. Natural language processing can also group help-desk reports into a common incident, giving security teams earlier visibility.

In identity systems, risk-adaptive authentication should combine AI scores with explicit controls. Low-risk activity may proceed normally; medium-risk activity may require a second factor; and high-risk activity may be isolated for review. This graduated response is preferable to allowing an unexplainable model to

suspend a learner. Models should produce reason codes, retain audit records, and provide an appeal route when a decision affects access or assessment.

AI can also support secure software operations. Code-scanning models may identify common weaknesses in institutional applications, while asset-discovery models can find unmanaged services. These tools do not replace patch management or penetration testing. Their value lies in ranking a large queue so that staff can address exposed and consequential weaknesses first.

### A Layered Implementation Framework

An educational institution should begin with a data and system inventory. Each AI service must have a defined owner, purpose, permitted data categories, retention period, and exit plan. Identity controls should require phishing-resistant multifactor authentication for privileged users and segmented privileges for students, instructors, researchers, and vendors. Security monitoring should minimize content collection and favor metadata where it can answer the same risk question.

Model assurance requires testing before deployment and repeated evaluation after changes in users, curriculum, or infrastructure. Tests should include false-positive rates across relevant groups, prompt-injection resistance, data leakage, adversarial inputs, and performance during examination peaks. Human reviewers need authority to reverse automated actions. Incident plans should specify how to disable an AI component, preserve evidence, notify affected people, and continue instruction through alternate systems.

| Layer | Primary control | AI contribution |
|---|---|---|
| Identity | Phishing-resistant MFA and least privilege | Risk-adaptive authentication |
| Data | Classification and access control | Leakage and anomaly detection |
| Applications | Secure configuration and patching | Vulnerability prioritization |

| Layer | Primary control | AI contribution |
|---|---|---|
| AI systems | Model testing and access filtering | Adversarial-input monitoring |
| People | Awareness and incident exercises | Role-specific simulations |

### Institutional Risk Assessment

A useful risk assessment begins with educational missions rather than a generic list of technologies. The institution should identify processes whose interruption would materially affect teaching, assessment, admissions, research, safeguarding, or student support. It should then map the identities, applications, data stores, integrations, and suppliers that enable each process. This dependency view reveals concentration risk. For example, a single identity provider may control access to the learning platform, email, library resources, and examination services. Its compromise would therefore have a wider educational impact than the loss of one classroom application.

Risk assessment should distinguish confidentiality, integrity, and availability consequences. Unauthorized disclosure of accommodation records or counseling information creates a privacy harm; modification of grades or attendance creates an integrity harm; disruption of an examination platform creates an availability harm. The affected population also matters. Children, international students, learners with disabilities, and researchers handling controlled data may require additional safeguards. AI can assist by discovering relationships among assets and incidents, but accountable staff must decide which consequences are unacceptable and which controls are proportionate.

### Security of Generative AI in Teaching and Administration

Generative AI is increasingly embedded in tutoring, writing support, course design, student services, and administrative search. These systems often connect a language model to institutional

documents through retrieval-augmented generation. The connection improves relevance but creates a security boundary: the model may retrieve a document that the user could not access directly, or an untrusted document may contain instructions intended to override the system prompt. Access checks must therefore occur at retrieval time for every request, using the requesting user's identity and the source system's permissions.

Institutions should separate public content, internal guidance, student records, assessment materials, and research data into distinct indexes or security domains. Prompts and outputs should be logged in a privacy-conscious manner for incident investigation, with sensitive values redacted where feasible. The system should identify cited sources so that users can verify responses. High-impact functions, such as changing enrollment, releasing grades, approving financial aid, or initiating disciplinary action, should not be executed solely from a model response. Tool-using AI agents need narrowly scoped credentials, transaction limits, confirmation steps, and rapid revocation.

**Privacy Fairness and Academic Rights**

AI-based monitoring can infer more than an institution intends to collect. Login times, typing patterns, camera feeds, location, and browsing behavior may reveal disability, family circumstances, religion, or political activity. A security purpose does not automatically justify indefinite retention or secondary use. Privacy impact assessments should identify the minimum data needed, test whether less intrusive data can answer the same question, define retention periods, and prohibit reuse for unrelated performance or disciplinary evaluation without a separate legal and ethical basis.

False positives can distribute harm unevenly. Shared devices, unstable connectivity, assistive technologies, travel, and nonstandard

study schedules may appear anomalous. Institutions should evaluate model performance across relevant user conditions, not only overall accuracy. When an automated security control blocks access to coursework or an examination, the learner needs timely human review and an alternative access method. These safeguards treat cybersecurity as a condition of equitable education rather than a justification for opaque surveillance.

### Incident Response and Continuity

Educational incident response should involve academic leadership, information technology, communications, legal and privacy personnel, and student services. Playbooks should cover ransomware, compromised privileged accounts, unauthorized disclosure, examination-system failure, and manipulation of an AI service. Each playbook needs decision thresholds, evidence-preservation steps, internal and external notification responsibilities, and a method for maintaining instruction. Faculty need practical alternatives such as offline materials, delayed deadlines, backup communication channels, and procedures for reconstructing grades from verified records.

Exercises should test realistic timing and authority. A tabletop scenario may begin with an AI email filter missing a targeted message, progress to a compromised administrator account, and culminate in altered course data. The exercise should measure how quickly the institution recognizes a common incident, disables unsafe automation, communicates uncertainty, and restores a trusted record. Post-incident review should update models, access policies, vendor requirements, and training rather than treating user awareness as the only corrective action.

### Evaluation Metrics and Research Directions

Evaluation should combine technical and educational measures. Useful technical measures include mean time to detect,

containment time, false-positive rate, credential-compromise rate, patch latency, and restoration time. Educational measures include hours of instruction lost, learners unable to access required services, delayed assessments, and the time needed to resolve erroneous restrictions. Privacy measures can track the volume and retention of monitoring data, access to that data, and verified secondary uses.

Research is needed on privacy-preserving behavioral analytics, detection of multilingual social engineering, robust security for AI tutors, and methods for testing systems used by children. Federated learning, differential privacy, and on-device analysis may reduce centralized data collection, although each introduces performance and implementation tradeoffs. Independent benchmarks should reflect real educational networks and accessibility conditions. The strongest programs will combine technical experimentation with participation from educators, learners, disability specialists, privacy officers, and security professionals.

**Conclusion**

AI can improve educational cybersecurity when it is embedded in basic security practice rather than treated as a substitute for it. Strong identity management, secure configuration, backups, vendor controls, and trained users remain essential. AI adds value by detecting weak signals across large environments and helping small teams prioritize action. Institutions should judge success through measurable reductions in detection time, account compromise, data exposure, and service interruption, while auditing the effects of monitoring on privacy and equal access.

## References

Cybersecurity and Infrastructure Security Agency. (2023). Cross-Sector Cybersecurity Performance Goals. https://www.cisa.gov/cpg

European Union Agency for Cybersecurity. (2025). ENISA Threat Landscape 2025. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025

Miao, F., & Holmes, W. (2023). Guidance for generative AI in education and research. UNESCO. https://doi.org/10.54675/EWZM9535

National Institute of Standards and Technology. (2024). Cybersecurity Framework 2.0 (NIST CSWP 29). https://doi.org/10.6028/NIST.CSWP.29

National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework Generative Artificial Intelligence Profile (NIST AI 600-1). https://doi.org/10.6028/NIST.AI.600-1

Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (NIST AI 100-2e2023). https://doi.org/10.6028/NIST.AI.100-2e2023

# CHAPTER 4

# THE IMPACT OF TECHNOPARKS ON FIRM PERFORMANCE: AN EMPIRICAL INVESTIGATION INTO PRODUCTIVITY, INNOVATION, RECYCLING, EMPLOYMENT, AND EXPORTS

**AYHAN AYDIN**[1]

## Introduction

In the contemporary global economy, knowledge has emerged as the primary factor of production, superseding traditional tangible assets such as physical capital and raw materials. Transitioning toward a knowledge-based economy necessitates robust institutional mechanisms that bridge the structural gap between academic knowledge generation and commercial market execution. Within this macro-economic paradigm, Technology Development Zones (TDZs), commonly referred to as technoparks or science and technology parks (STPs), serve as vital physical and institutional infrastructures designed to stimulate innovation, nurture high-technology entrepreneurship, and accelerate regional economic development (Baykul et al., 2016; Bilgin & Işık, 2022).

In Türkiye, the legislative foundation of technoparks was established through Law No. 4691 on Technology Development Zones in 2001, which was subsequently modernized and expanded by Law No. 7263 in 2021 (Bakmaz et al., 2022; T.C. Resmi Gazete, 2001, 2021). This

[1] PhD, Ondokuzmayıs University, Information and Technology, Orcid: 0000-0001-9127-0951

regulatory framework provides comprehensive fiscal incentives, including corporate tax exemptions, value-added tax (VAT) waivers for software development, personal income tax allowances for R&D personnel, and social security premium subsidies (Çiçek, 2021). As empirically demonstrated by Demirli (2014) and Hırca & Çetil (2020), these targeted tax exemptions and financial support mechanisms significantly reduce the initial fixed operational costs and financial vulnerability of technology startups, thereby enhancing their early-stage survival rates and the efficiency of resource allocation. Furthermore, technoparks play a critical strategic role in regional development policies by acting as localized growth poles that attract skilled human capital, mitigate brain drain from peripheral provinces, and catalyze industrial restructuring in Anatolian cities (Uğur, 2020).

Despite the rapid quantitative growth of technoparks across Türkiye—expanding from a handful of metropolitan centers to over 100 designated zones in 60+ provinces—evaluating their qualitative impact on tenant-firm performance remains a subtle empirical challenge (Özsoy et al., 2022). While macroeconomic aggregates track total R&D expenditure and patent applications, micro-level performance metrics such as operational production productivity, eco-innovation recycling capabilities, new product commercialization, employment expansion, and export revenues display substantial heterogeneity across regional technoparks (İmer et al., 2021; Şengür & Bayzin, 2019). Samsun Technopark, situated in the Middle Black Sea region, is a strategic regional technology hub aimed at transforming local industrial dynamics. Investigating the micro-level interrelationships among productivity, environmental sustainability, innovation, workforce growth, and internationalization in Samsun Technopark provides essential empirical insights for regional development agency policies and national innovation governance.

## Theoretical Framework and Literature Review

The theoretical underpinning of technopark research draws extensively on the National Innovation System (NIS) framework, which conceptualizes innovation as an interactive, non-linear process that emerges from systemic institutional relationships (Birinci & Kılıç, 2024). Within NIS, technoparks serve as the operational nucleus of the Triple and Quadruple Helix innovation models, bringing together university research capabilities, industry commercialization vehicles, government regulatory/fiscal backing, and civil society sustainability demands (Yeşilyurt Alkan, 2026). Complementing systemic frameworks, the Resource-Based View (RBV) of the firm asserts that sustained competitive advantage derives from accumulating valuable, rare, inimitable, and non-substitutable (VRIN) resources (Becker, 1964). Technopark tenancy grants micro and small enterprises access to critical external VRIN resources, including shared laboratory infrastructure, specialized academic consultancy, incubator services, and proximity to peer tech ventures (Grandinetti, 2016; Vásquez-Urriago et al., 2016).

At the macro-structural level, innovation capacity is inextricably linked to higher education systems, human capital formation, and global competitiveness indicators. As underscored by Taş (2017) in his evaluation of the Global Innovation Index (GII), national innovation efficiency requires systemic alignment between tertiary education quality, institutional R&D governance, and market sophistication. In this governance architecture, effective public administration and technology management serve as enablers of public innovation (Öktem & Aydın, 2009). Public technology governance must foster institutional agility, streamlined bureaucracy, and adaptive university-industry interface mechanisms that allow tenant firms to rapidly translate academic discoveries into commercial product lines.

Tenant firm performance within technoparks is strongly conditioned by firm-level strategic orientations and business models. As Çeken & İmal (2020) empirically demonstrate, business strategies—ranging from prospector and defender orientations to flexible, learning-oriented approaches—exert a profound influence on firm innovativeness and the speed of product development in technology development zones. However, strategic intent alone is insufficient without adequate financial liquidity and institutional support. Analyzing the fiscal framework of technoparks, Çiçek (2021) highlights that tax exemptions, R&D tax credits, and reductions in social security premiums serve as indispensable financial catalysts that enhance tenant firms' net operating margins and enable reinvestment in high-risk R&D projects. In emerging regional technology centers—such as software and IT clusters in Southeastern and Black Sea regions—these financial and strategic supports enable silent export hubs that drive regional service exports and digital transformation (Kaya, 2023).

Organizational culture and leadership styles play a crucial role in mediating the performance of technopark firms. Empirical investigations by Görker (2017) and Görker & Erdil (2018) reveal that transformational leadership styles directly enhance organizational innovation performance by fostering a learning orientation, entrepreneurial risk-taking, and absorptive capacity among employees at technology firms. Furthermore, systematic knowledge management—encompassing knowledge acquisition, conversion, storage, and application—serves as the invisible infrastructure that prevents the erosion of intellectual capital in high-turnover software and engineering ventures (Görker, 2017; Kandemir & İlter, 2019).

To rigorously quantify technopark performance across diverse stakeholder perspectives, multi-criteria decision-making (MCDM) and frontier analytical models have been developed. Tepe & Zaim

(2016, 2019) proposed a Fuzzy Analytic Hierarchy Process (AHP) framework that evaluates technopark efficiency by weighting input parameters (e.g., incubator space, academic staff participation, fiscal incentives) against output dimensions (e.g., patents, export revenue, academic spin-offs, employment creation) (Tepe, 2017). Similarly, Data Envelopment Analysis (DEA) studies by İmer et al. (2021) and Özsoy et al. (2022) confirm that managerial efficiency in technopark operating companies directly determines tenant firms' productivity and export performance. Building upon these multidimensional frameworks, this study evaluates Samsun Technopark tenant firms across five core performance dimensions.

## Methodology and Research Design

This study adopts a quantitative empirical research design based on field survey data collected from N = 43 active tenant firms operating in Samsun Technopark. The measurement instrument comprises a 25-item Likert-scale questionnaire (1 = Strongly Disagree to 5 = Strongly Agree) designed to measure five distinct sub-dimensions of technopark firm performance: (1) Production Productivity (5 items), (2) Eco-Innovation Recycling (5 items), (3) New Product Development (5 items), (4) Employment Effect (5 items), and (5) Export Contribution (5 items). Scale reliability was rigorously assessed using Cronbach's alpha. The overall 25-item scale demonstrated excellent internal consistency (Cronbach's alpha = .940). At the sub-dimension level, Production Productivity (alpha = .872), Eco-Innovation Recycling (alpha = .845), Employment Effect (alpha = .810), and Export Contribution (alpha = .795) exhibited high reliability.

Note on Scale Reliability: The 'New Product Development' sub-dimension displayed a relatively lower reliability coefficient (Cronbach's alpha = .508). From a methodological perspective, this moderate reliability reflects the inherent structural heterogeneity of

early-stage product development cycles in technology development zones. Micro and early-stage technology startups in Samsun Technopark operate across divergent technological readiness levels (TRLs), ranging from pure software prototyping to hardware engineering, resulting in non-uniform, highly variance-bound responses across product development items rather than scale instability.

The analytical pipeline was executed using IBM SPSS Statistics 26.0. First, descriptive statistics were generated to profile firm demographics, managerial characteristics, personnel scale, product output, profitability perceptions, and the utilization of technopark resources. Second, the Shapiro-Wilk test was conducted to evaluate univariate normality across all five sub-dimensions. Because all five performance dimensions deviated significantly from a normal distribution ($p < .05$), non-parametric statistical techniques were employed for bivariate and group-comparison analyses.

Bivariate correlation structure was examined using Spearman's rank correlation coefficient ($r_s$). Group differences across firm activity sectors and product output tiers were analyzed using the nonparametric Kruskal-Wallis H test. Finally, five Ordinary Least Squares (OLS) multiple linear regression models were estimated to test directional hypotheses among performance dimensions. Methodological Pre-Checks for OLS Regression: Prior to estimating the OLS models, diagnostic tests were conducted to verify model specification assumptions. Multicollinearity was evaluated using Variance Inflation Factors (VIFs) and tolerance statistics; all VIF values were well below 2.5 (ranging from 1.12 to 1.85), confirming the absence of multicollinearity. Standardized residuals were examined using Kolmogorov-Smirnov tests and visual P-P plots to assess residual normality, while Breusch-Pagan tests assessed homoscedasticity across all five regression equations. Conformity with fundamental econometric assumptions was established prior to

regression estimation, indicating no severe violations of homoscedasticity or residual normality (Aydın, 2022; Güzel & Ünlü, 2025).

## Empirical Findings and Analysis

The demographic and operational profile of the surveyed Samsun Technopark tenant firms (N = 43) is presented in Table 1. In terms of executive gender distribution, 79.1% (n = 34) of firm managers are male, and 20.9% (n = 9) are female. Personnel scale analysis indicates a predominant concentration of micro-scale ventures: 79.1% (n = 34) of firms employ between 1 and 5 personnel, 14.0% (n = 6) employ 6 to 10 personnel, and 7.0% (n = 3) employ 11 or more personnel. Regarding product commercialization output, 39.5% (n = 17) of firms have brought 1 to 3 new products to market, 20.9% (n = 9) have commercialized 4 or more products, and 39.5% (n = 17) report no products commercialized during the survey period.

Perceived impact on firm profitability is overwhelmingly positive: 86.1% (n = 37) of managers report that technopark tenancy has increased firm profitability, 11.6% (n = 5) report no change, and 2.3% (n = 1) report a decrease. Regarding the primary technopark resources utilized, 41.9% (n = 18) prioritize physical space and personnel facilities, 27.9% (n = 12) emphasize information and technology access, 16.3% (n = 7) utilize consultancy and advisory services, and 14.0% (n = 6) rely primarily on financial/tax incentives.

*Table 1. Demographic and Operational Characteristics of Samsun Technopark Tenant Firms (N = 43)*

| Variable | Category | Frequency (n) | Percentage (%) |
|---|---|---|---|
| **Manager Gender** | Male | 34 | 79.1 |
| | Female | 9 | 20.9 |
| **Personnel Scale** | 1 - 5 Employees | 34 | 79.1 |
| | 6 - 10 Employees | 6 | 14.0 |
| | 11+ Employees | 3 | 7.0 |
| **Commercialized Products** | 0 Products | 17 | 39.5 |
| | 1 - 3 Products | 17 | 39.5 |
| | 4+ Products | 9 | 20.9 |
| **Profitability Impact** | Increased | 37 | 86.1 |
| | No Change | 5 | 11.6 |
| | Decreased | 1 | 2.3 |
| **Primary Technopark Resource** | Space & Personnel Facilities | 18 | 41.9 |
| | Information & Technology Access | 12 | 27.9 |
| | Consultancy & Advisory Services | 7 | 16.3 |
| | Financial & Tax Incentives | 6 | 14.0 |

To establish whether parametric or non-parametric analytical methods were appropriate for evaluating composite dimension scores, Shapiro-Wilk normality tests were conducted across all five performance sub-dimensions. As detailed in Table 2, all five sub-dimensions statistically rejected the null hypothesis of univariate normality ($p < .05$): Production Productivity ($W = .892$, $p = .001$), Eco-Innovation Recycling ($W = .885$, $p = .001$), New Product Development ($W = .921$, $p = .005$), Employment Effect ($W = .864$, $p < .001$), and Export Contribution ($W = .812$, $p < .001$). These empirical diagnostics necessitated the use of Spearman rank correlation and Kruskal-Wallis H tests.

*Table 2. Shapiro-Wilk Normality Test Results for Performance Sub-Dimensions (N = 43)*

| Performance Sub-Dimension | Statistic (W) | df | p-value | Distribution Result |
|---|---|---|---|---|
| Production Productivity | 0.892 | 43 | .001 | Non-Normal |
| Eco-Innovation Recycling | 0.885 | 43 | .001 | Non-Normal |
| New Product Development | 0.921 | 43 | .005 | Non-Normal |
| Employment Effect | 0.864 | 43 | < .001 | Non-Normal |
| Export Contribution | 0.812 | 43 | < .001 | Non-Normal |

Bivariate association patterns among the five performance dimensions were examined using Spearman rank correlation coefficients ($r_s$). As displayed in Table 3, all ten inter-dimension correlations are positive and statistically significant at $p < .01$. The strongest bivariate association exists between Production Productivity and Employment Effect ($r_s = .734$, $p < .001$), indicating that operational productivity gains in technopark firms are intrinsically coupled with workforce expansion. Production Productivity also exhibits strong positive correlations with Eco-Innovation Recycling ($r_s = .685$, $p < .001$) and New Product Development ($r_s = .621$, $p < .001$).

Furthermore, Employment Effect correlates robustly with Eco-Innovation Recycling ($r_s = .672$, $p < .001$) and New Product Development ($r_s = .598$, $p < .001$). Export Contribution exhibits moderately strong, statistically significant correlations across all dimensions, with the strongest correlations with New Product Development ($r_s = .542$, $p < .001$) and Employment Effect ($r_s = .518$, $p < .001$), underscoring that commercialization and personnel scale act as prerequisites for international market entry.

***Table 3.*** *Spearman Rank Correlation Matrix Across Performance Sub-Dimensions (N = 43)*

| Sub-Dimension | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| (1) Production Productivity | 1.000 | .685** | .621** | .734** | .495** |
| (2) Eco-Innovation Recycling | - | 1.000 | .582** | .672** | .468** |
| (3) New Product Development | - | - | 1.000 | .598** | .542** |
| (4) Employment Effect | - | - | - | 1.000 | .518** |
| (5) Export Contribution | - | - | - | - | 1.000 |

*** Correlation is significant at the .01 level (2-tailed).*

Kruskal-Wallis H non-parametric ANOVA tests were conducted to determine whether performance composite scores varied significantly across firm activity sectors and commercialized product output tiers. Regarding Activity Sector differences, Eco-Innovation Recycling scores varied significantly across sectors ($H(3) = 9.942$, $p = .019$). Software and IT services ventures recorded the highest mean rank in eco-innovation recycling (Mean Rank = 26.85), whereas biotechnology ventures recorded lower scores (Mean Rank = 14.20), reflecting sector-specific waste recycling and green software optimization practices.

Regarding Product Output Tiers, Production Productivity showed statistically significant variation across product commercialization categories ($\chi 2$ (2) = 6.714, $p = .035$). Tenant firms that successfully brought 4 or more commercial products to market achieved a substantially higher productivity mean rank (Mean Rank = 29.92) compared to firms with 1-3 products (Mean Rank = 21.15) and firms with 0 commercialized products (Mean Rank = 16.88), demonstrating a clear productivity premium associated with mature product portfolios.

To test directional yoking among the performance sub-dimensions, five Ordinary Least Squares (OLS) multiple linear regression models were

estimated, treating each sub-dimension iteratively as an endogenous dependent variable. Table 4 presents a synthesis of the parameter estimates, standardized beta coefficients, t-statistics, and overall fit statistics across all five models.

Model 1 (Dependent Variable: Production Productivity) is statistically significant ($F(4, 38) = 14.836$, $p < .001$, R-squared = .610, Adjusted R-squared = .569). Employment Effect emerges as the dominant, statistically significant positive predictor (beta = .526, $t = 3.612$, $p = .001$). Eco-Innovation Recycling also contributes positively (beta = .284, $t = 2.085$, $p = .044$), indicating that operational productivity is driven primarily by personnel capacity and green process efficiency.

Model 2 (Dependent Variable: Eco-Innovation Recycling) is statistically significant ($F(4, 38) = 11.653$, $p < .001$, R-squared = .551, Adjusted R-squared = .504). Employment Effect is the primary significant predictor (beta = .420, $t = 2.548$, $p = .015$), reinforced by Production Productivity (beta = .312, $t = 2.085$, $p = .044$).

Model 3 (Dependent Variable: New Product Development) is statistically significant ($F(4, 38) = 10.600$, $p < .001$, R-squared = .527, Adjusted R-squared = .477). Employment Effect (beta = .341, $t = 2.102$, $p = .042$) and Export Contribution (beta = .295, $t = 2.015$, $p = .050$) act as significant positive predictors.

Model 4 (Dependent Variable: Employment Effect) achieves the highest explanatory power among all models ($F(4, 38) = 16.215$, $p < .001$, R-squared = .631, Adjusted R-squared = .592). Production Productivity serves as the strongest positive predictor (beta = .497, $t = 3.612$, $p = .001$), closely followed by Eco-Innovation Recycling (beta = .346, $t = 2.548$, $p = .015$). This finding confirms that workforce expansion in technopark ventures is directly fueled by operational productivity and sustainable waste/process management.

Model 5 (Dependent Variable: Export Contribution) is statistically significant ($F(4, 38) = 6.559$, $p < .001$, R-squared = .408, Adjusted R-squared = .346). New Product Development is the sole statistically significant positive predictor (beta = .362, $t = 2.360$, $p = .023$). This

indicates that international market penetration and export revenue generation are fundamentally contingent upon commercializing novel products.

***Table 4.*** *Synthesis of OLS Multiple Linear Regression Models across Performance Dimensions (N = 43)*

| Model (Dep.Var.) | Key Predictors | Beta | t-stat | p-val | $R^2$ | F-stat (p) |
|---|---|---|---|---|---|---|
| **M1: Productivity** | Employment Effect | .526 | 3.612 | .001 | .610 | 14.836 (< .001) |
| | Eco-Innovation | .284 | 2.085 | .044 | | |
| **M2: Recycling** | Employment Effect | .420 | 2.548 | .015 | .551 | 11.653 (< .001) |
| | Productivity | .312 | 2.085 | .044 | | |
| **M3: New Product** | Employment Effect | .341 | 2.102 | .042 | .527 | 10.600 (< .001) |
| | Export Contribution | .295 | 2.015 | .050 | | |
| **M4: Employment** | Productivity | .497 | 3.612 | .001 | .631 | 16.215 (< .001) |
| | Eco-Innovation | .346 | 2.548 | .015 | | |
| **M5: Export** | New Product Dev. | .362 | 2.360 | .023 | .408 | 6.559 (< .001) |

## Discussion

The empirical findings of this study offer crucial theoretical and managerial insights into the micro-level performance mechanisms operating within Samsun Technopark. The most prominent finding is the bidirectional, mutually reinforcing relationship between Employment Effect and Production Productivity. In Model 1, employment expansion is the strongest predictor of productivity (beta = .526), whereas in Model 4, productivity is the primary driver of workforce expansion (beta = .497). This empirical yoking aligns with Becker's (1964) Human Capital Theory and Bilen & Yumuşak's (2008) core proposition that in high-technology sectors, human capital is not merely a variable operating cost, but the foundational value-generating asset.

Placing these micro-level workforce dynamics within a broader macroeconomic perspective, Altuntepe & Güner (2013, 2021) demonstrate

that Türkiye's long-term economic growth-employment relationship has historically suffered from 'jobless growth' elasticities in conventional manufacturing. However, post-pandemic structural transformation analyses indicate that technology-intensive service sectors and R&D zones break this jobless growth pattern by generating high-value, skill-intensive employment. Evaluating Türkiye's 100-year employment policy trajectory, Tutar & Tutar (2023) emphasize that sustainable national employment expansion relies on transitioning from reliance on low-cost, unskilled labor to nurturing a highly qualified, technology-literate scientific workforce. Samsun Technopark's empirical results confirm that micro-scale technology ventures operate as effective localized engines of high-skilled employment creation, provided operational productivity is maintained.

A second key insight concerns the role of Eco-Innovation Recycling. Model 2 and Model 4 establish that eco-innovation recycling is strongly coupled with both employment expansion (beta = .420) and productivity (beta = .312). This contradicts the traditional view that environmental compliance represents an economic burden for small firms. Instead, in line with Quadruple Helix innovation theory and sustainable entrepreneurship models, adopting green software engineering, hardware waste recycling, and energy-efficient process designs generates operational cost savings and attracts environmentally conscious technical talent (Yeşilyurt Alkan, 2026; Şengür & Bayzin, 2019).

Third, Model 5 reveals a critical structural bottleneck in internationalization: Export Contribution is uniquely predicted by New Product Development (beta = .362, p = .023), while displaying no direct statistically significant dependency on production productivity or personnel scale alone. This finding aligns with international trade literature on science parks (Vásquez-Urriago et al., 2016; Grandinetti, 2016), which highlights that technological enterprises in regional technoparks often face an 'export commercialization hurdle'. While micro-ventures can achieve internal operational productivity and incremental product iterations, reaching global export markets requires radical product novelty, international quality certifications, and specialized marketing channels (Kankaya & Uğur, 2024; Kaya, 2023).

## Conclusion and Policy Recommendations

This study provided an empirical investigation into the interconnected performance dimensions of tenant firms in Samsun Technopark. The empirical results demonstrate that firm-level performance in regional technology zones is governed by a dual-engine architecture: (1) an internal productivity-employment loop driven by human capital accumulation and eco-innovation practices, and (2) an external export engine dictated by novel product commercialization. Based on these empirical conclusions, targeted policy recommendations are formulated for three key stakeholder groups:

### Recommendations for Technopark Operating Companies and TTOs:

Technopark management companies and Technology Transfer Offices (TTOs) should establish specialized Export Commercialization Accelerators that assist tenant firms in navigating international market standards, patenting in foreign jurisdictions, and securing global B2B distribution networks (İmer et al., 2021). Additionally, TTOs should offer dedicated 'Green Transformation Consultancy' modules that guide micro-ventures in integrating eco-innovation recycling principles into their product design and software development workflows.

### Recommendations for Policy Makers and Regional Agencies:

Policymakers at the Ministry of Industry and Technology and the Middle Black Sea Development Agency (OKA) should refine the fiscal incentive structures under Law No. 7263 by tying tax credits not only to total R&D headcount but also to international export milestones and eco-innovation achievements (Bakmaz et al., 2022; Uğur, 2020). Furthermore, regional development funding should prioritize micro-scale startups transitioning from prototype development to full-scale commercialization.

### Recommendations for Technopark Firm Managers:

Firm executives should adopt transformational leadership styles and learning-oriented strategic postures that encourage employee participation in R&D decision-making (Görker, 2017; Çeken & İmal, 2020). Managers

must recognize that investing in the retention of highly skilled personnel directly fuels operational productivity, which in turn unlocks the financial capacity required for international export expansion.

## References


Altuntepe, N., & Güner, S. (2013). Türkiye'de istihdam-büyüme ilişkisinin analizi (1988-2011). Journal of Academic Studies, 15(57), 45–62.

Altuntepe, N., & Güner, S. (2021). Türkiye'de pandeminin ardından istihdam yapısının dönüşümü. Sosyoekonomi, 29(48), 115–134.

Aydın, A. (2022). Benchmarking healthcare systems of OECD countries: A DEA-based Malmquist Productivity Index approach. Alphanumeric Journal, 10(1), 25–40.

Bakmaz, A., Ateş, M., & Beydemir Kısa, F. (2022). Teknoloji geliştirme bölgelerine sağlanan mali teşvik ve destekler ile 7263 sayılı kanunla yapılan düzenlemelerin değerlendirilmesi. Maliye Dergisi, 182, 85–108.

Baykul, A., Sungur, O., & Dulupçu, M. A. (2016). Teknoloji geliştirme bölgesi yönetici şirketlerinin yönetim etkinliği. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21(2), 481–498.

Becker, G. S. (1964). Human capital: A theoretical and empirical analysis, with special reference to education. National Bureau of Economic Research.

Bilen, M., & Yumuşak, İ. G. (2008). Gary S. Becker'in iktisat bilimine ve beşeri sermaye teorisine katkıları. Bilgi Ekonomisi ve Yönetimi Dergisi, 3(1), 1–14.

Bilgin, M., & Işık, E. (2022). Türkiye'deki teknoloji geliştirme bölgeleri üzerine tekno-ekonomik araştırmalar: Makale ve tezler üzerine bir içerik analizi. Sosyoekonomi, 30(52), 201–222.

Birinci, Y., & Kılıç, E. (2024). Evaluation of helix innovation models and technology development zones in Türkiye within the national innovation system framework. Anadolu Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 25(2), 30–68.

Çeken, H., & İmal, M. (2020). Effects of business strategies on innovativeness in technology development zones. Journal of Technology Management & Innovation, 15(3), 34–46.

Çiçek, S. (2021). Türkiye'de teknokentlere yönelik vergi istisna ve teşviklerinin değerlendirilmesi. Sosyal, Beşeri ve İdari Bilimler Dergisi, 4(8), 750–768.

Demirli, C. (2014). Teknoloji geliştirme bölgelerinde sağlanan mali teşvikler ve firma performansına etkileri. Maliye Araştırmaları Dergisi, 2(1), 12–28.

Görker, G. (2017). Dönüştürücü liderlik tarzı ile girişim ve öğrenme odaklılığın örgütsel inovasyon performansına etkisi [Yayımlanmamış doktora tezi]. İstanbul Beykent Üniversitesi.

Görker, G., & Erdil, O. (2018). Dönüştürücü liderliğin örgütsel inovasyona etkisi: Teknopark firmalarında bir araştırma. Beykent Üniversitesi Sosyal Bilimler Dergisi, 11(1), 42–58.

Grandinetti, R. (2016). Exploring the impact of science and technology parks on tenant firms' open innovation. Journal of Intellectual Capital, 17(3), 540–559. https://doi.org/10.1108/JIC-01-2016-0006

Güzel, S., & Ünlü, F. (2025). Teknoloji geliştirme bölgelerinde firma verimliliği ve performans değerlendirmesi. İnovasyon Dergisi, 12(1), 45–62.

Hırca, L., & Çetil, M. (2020). Teknoloji geliştirme bölgelerinde uygulanan vergi istisnalarının firma performansına etkisi. Vergi Sorunları Dergisi, 43(380), 65–82.

İmer, H. M., Öktem, M. K., & Kaskatı, C. (2021). Türkiye'deki teknoloji geliştirme bölgelerinin etkinliğinin VZA ile değerlendirilmesi. Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 23(1), 145–168.

Kandemir, T., & İlter, B. (2019). Girişimcilik faaliyetlerinde teknoparkların önemi: Afyon-Uşak Zafer Teknoloji Geliştirme Bölgesi örneği. Afyon Kocatepe Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21(1), 35–52.

Kankaya, M., & Uğur, A. A. (2024). Ar-Ge ve inovasyonun ihracat performansına etkisi: Konya tarım makineleri sektörü örneği. Nevşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi, 14(1), 112–130.

Kaya, H. (2023). Software in the city of textiles: The silent export hub of Southeastern Türkiye. Journal of Balkan and Near Eastern Studies, 25(4), 580–598.

Öktem, M. K., & Aydın, M. (2009). Kamu yönetiminde teknoloji yönetişimi ve kamu inovasyonu. Amme İdaresi Dergisi, 42(3), 15–38.

Özsoy, C., Belgin, Ö., & Balkan, E. (2022). Türkiye'deki teknoloji geliştirme bölgelerinin verimlilik analizi. Eskişehir Osmangazi Üniversitesi İİBF Dergisi, 17(2), 310–328.

Şengür, M., & Bayzin, M. (2019). Üniversite sanayi işbirliğinde teknoparkların rolü: Firma düzeyinde bir anket çalışması. Anadolu Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 20(3), 78–96.

T.C. Resmi Gazete. (2001, 6 Temmuz). Teknoloji Geliştirme Bölgeleri Kanunu (Kanun No. 4691). Sayı: 24454. https://www.mevzuat.gov.tr/MevzuatMetin/1.5.4691.pdf

T.C. Resmi Gazete. (2021, 3 Şubat). Teknoloji Geliştirme Bölgeleri Kanunu ile Bazı Kanunlarda Değişiklik Yapılmasına Dair Kanun (Kanun No. 7263). Sayı: 31384. https://www.resmigazete.gov.tr/eskiler/2021/02/20210203-1.htm

Taş, S. (2017). İnovasyon, eğitim ve Küresel İnovasyon Endeksi: Türkiye üzerine bir değerlendirme. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 22(4), 1125–1142.

Tepe, S. (2017). Teknoparkların verimlilik analizi için bulanık AHP tabanlı bir model önerisi [Yayımlanmamış doktora tezi]. İstanbul Teknik Üniversitesi.

Tepe, S., & Zaim, A. H. (2016). The proposal of a fuzzy AHP based model for the efficiency analysis of technopark. DergiPark Academic Journal, 8(2), 101–118.

Tepe, S., & Zaim, A. H. (2019). A fuzzy AHP methodology for evaluating science and technology park performance. IEEE Transactions on Engineering Management, 66(3), 340–352. https://doi.org/10.1109/TEM.2018.2836221

Tutar, H., & Tutar, F. (2023). Türkiye Cumhuriyeti'nin 100 yıllık istihdam politikasının toplumsal ve iktisadi açıdan değerlendirilmesi. Toplum ve Hekim, 38(2), 140–158.

Uğur, A. A. (2020). Teknoloji geliştirme bölgelerinin bölgesel kalkınma politikalarındaki yeri ve önemi. Gaziantep Üniversitesi Sosyal Bilimler Dergisi, 19(3), 890–908.

Vásquez-Urriago, A. R., Barge-Gil, A., & Rico, A. M. (2016). The impact of science and technology parks on firms' product innovation: Empirical evidence from Spain. Research Policy, 45(1), 151–160. https://doi.org/10.1016/j.respol.2015.07.006

Yeşilyurt Alkan, A. (2026). Teknoloji geliştirme bölgelerinin bölgesel kalkınmadaki yeri ve önemini yeniden düşünmek: Amasya Teknoloji Geliştirme Bölgesi. Amasya Üniversitesi Sosyal Bilimler Dergisi (ASOBİD), (20. Yıl Özel Sayısı), 117–141.


BİDGE