ISSN: 2821-1553 htpps://www.jadm.khadu.ir

فصلنامه علمی ((دفاع هوافضایی))

دوره ۲، شماره ۳ ، آذر ۱۴۰۲

مقاله پژوهشی

پیش‌بینی حرکت آتی اهداف متخاصم مبتنی بر شبکه عصبی مصنوعی بر روی صفحه نمایش رادار

حمید محسنی[۱]، مهدی توکلی[۲]، محمد فرهمند راد[۳]، داوود نظری[۴]، حمید بلوچستانی[۵]

۱. مدرس دانشکده مهندسی برق، دانشگاه پدافند هوایی خاتم الانبیاء(ص)، تهران، ایران.

۲ و ۳ و ۴. مدرس دانشکده مهندسی برق، دانشگاه پدافند هوایی خاتم الانبیاء(ص)، تهران، ایران.

۵. کارشناسی ارشد، دانشکده مهندسی برق، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

تاریخ پذیرش: ۱۴۰۲/۱۱/ ۱۰

تاریخ دریافت: ۱۴۰۲/۰۶/۰۱

کلمات کلیدی:

آموزش، پیش‌بینی، شبکه عصبی، شبیه‌ساز، کتابخانه Encog

نویسنده مسئول:

دکتر حمید محسنی

ایمیل:

mohseny.ha@gmail.com

چکیده

با پیشرفت در فناوری رادار، آموزش اپراتورها برای تجزیه و تحلیل مؤثر و پاسخگویی به تهدیدات بالقوه بسیار مهم است. این تحقیق یک شبیه‌ساز نمایشگر رادار PPI را ارائه می‌دهد که با استفاده از C# و پلتفرم WPF پیاده سازی شده است. این شبیه‌ساز یک شبکه عصبی را که توسط کتابخانه Encog پشتیبانی می‌شود، با نمایشگر رادار ادغام می‌کند تا مکان‌های حمله و مسیرهای آتی اهداف شبیه‌سازی شده را پیش‌بینی کند. شبکه عصبی پیاده‌سازی شده دارای یک معماری چهار لایه است که به اپراتورها اجازه می‌دهد تا پارامترهایی متعدد شبکه را متناسب با نیاز خود تنظیم کنند. این شبیه‌ساز با دارا بودن قابلیت امکان تغییر داده‌های آموزشی از طریق فایل‌های اکسل، تعامل با کاربر را افزایش داده و همچنین امکان افزودن یا حذف مکان‌های هدف نیز فراهم شده است. این امکان زمینه را برای سناریوهای واقعی‌تر فراهم می‌کند. شبیه‌ساز پیاده‌سازی شده به دانشجویان دانشگاه پدافند هوایی خاتم الانبیا (ص) این امکان را می‌دهد تا مهارت‌های خود را در تجزیه و تحلیل داده‌های راداری تقویت کنند و با یک محیط یادگیری جامع و پیشرفته نمایشگرهای نسل جدید رادار مجهز به فناوری شبکه عصبی آشنا شوند.

استناد به مقاله: حمید محسنی، مهدی توکلی، محمد فرهمندراد، داوود نظری و حمید بلوچستانی. پیش‌بینی حرکت آتی اهداف متخاصم مبتنی‌بر شبکه عصبی مصنوعی برروی صفحه نمایش رادار. مجله علمی پژوهشی دفاع هوافضایی دوره ۲، شماره ۳، آذر۱۴۰۲

ISSN:2821-1588 htpps://www.jadm.khadu.ir

Journal of Aerospace Defense
Vol. 2, No, 3, 1402

Research Paper

Predicting the future movement of hostile targets based on artificial neural network on the radar screen

H. Mohseni[1], M. Tavakoli[2], M. Farahmandrad[3], D. Nazari[4], H. Balouchestani[5]

1. Corresponding Author, Faculty of Electrical Engineering, Khatam Al-Anbia Air Defense University, Tehran, Iran.
2, 3, 4. Faculty of Electrical Engineering, Khatam Al-Anbia Air Defense University, Tehran, Iran.
5. Master's Degree, Electrical Engineering Faculty, Kashan University, Kashan, Iran.

Article Information

Accepted: 1402/11/10

Recceived: 1402/06/01

Keywords:

Encog Library, Neural Network, Prediction, Simulator, Training

Corresponding anuthor:
Dr. Hamid Mohseni
Email:
mohseny.ha@gmail.com

Abstract

As radar technology advances, training operators to effectively analyze and respond to potential threats is critical. This research presents a PPI radar display simulator implemented using C# and WPF platform. This simulator integrates a neural network powered by the Encog library with a radar display to predict attack locations and future trajectories of simulated targets. The implemented neural network has a four-layer architecture that allows operators to adjust multiple network parameters to suit their needs. With the ability to change training data through Excel files, this simulator has increased interaction with the user, and it is also possible to add or remove target locations. This allows for more realistic scenarios. The implemented simulator allows the operators of Khatam al-Anbia Air Defense University to strengthen their skills in radar data analysis and familiarize themselves with a comprehensive and advanced learning environment of new generation radar displays equipped with neural network technology.

HOW TO CITE: H. Mohseni, M. Tavakoli, M. Farahmandrad, D. Nazari, H. Balouchestani. Predicting the future movement of hostile targets based on artificial neural network on the radar screen. Journal of Aerospace Defense, Vol. 2, No 3, 1402.

۱-مقدمه

در سال‌های اخیر، شبکه‌های هوشمند نقش بسزایی در زمینه‌های مختلف نظامی ایفا کرده و به مرور جایگزین اپراتورهای انسانی در سیستم‌های نوین نظامی می‌شوند. ارزیابی تهدید اهداف پرنده در سامانه‌های فرماندهی و کنترل توسط اپراتورهای خبره بر اساس دانش و تجربه انجام می‌پذیرد. تجزیه و تحلیل اطلاعات ورودی دریافتی از سیستم‌های تلفیق داده امری بسیار دشوار است که نیازمند تصمیم‌گیری‌های پیچیده‌ای می‌باشد. توانایی بالا و دقت سیستم‌های هوشمند به منظور پیش‌بینی تهدید اهداف پرنده بر اساس پارامترهای مختلف دریافتی، کمک شایانی در تصمیم‌گیری نهایی می‌کند [۱]. در حوزه پدافند هوایی، توانایی شناسایی و ارزیابی دقیق تهدیدات بالقوه بسیار مهم است. همانطور که فناوری به تکامل خود ادامه می‌دهد، نیاز به راه‌حل‌های آموزشی نوآورانه که اپراتورها را به مهارت‌هایی برای هدایت پیچیدگی‌های سیستم‌های راداری مدرن مجهز می‌کنند، افزایش می‌یابند [۲]. روش‌های متعددی امروزه به منظور بهبود تصمیم‌گیری و پیش‌بینی در حوزه پدافند به کار گرفته می‌شود. به طور کلی می‌توان گفت تصمیم‌گیری شبکه‌های هوشمند به دو گروه تقسیم‌بندی می‌شوند. **گروه اول** شامل دسته‌بندی و تقسیم اهداف به گروه‌های معین [۳و۴] و **گروه دوم** به ردیابی و پیش‌بینی مسیر آتی اهداف می‌پردازد. ردیابی هدف، یک موضوع تحقیقاتی اساسی و فعال در بسیاری از زمینه‌ها مانند رادار، ناوبری و سیستم‌های رانندگی هوشمند است [۵]. هدف اصلی ردیابی اهداف تخمین دقیق وضعیت هدف در فضای حرکت است و می‌تواند با استفاده از شبکه عصبی و یا منطق فازی انجام شود [۶].

این تحقیق نیز به بررسی پژوهشی پیشگامانه می‌پردازد که نه تنها شبیه‌ساز نمایشگر نشانگر موقعیت طرح رادار (PPI)[1] را ارائه می‌کند، بلکه از فناوری شبکه عصبی برای پیش‌بینی مکان‌های حمله و مسیرهای آتی اهداف استفاده می‌کند. به منظور پیاده‌سازی شبیه‌ساز از زبان #C و پلت‌فرم WPF[2] در نرم افزار ویژوال استودیو که مزایای قابل توجهی را ارائه می‌دهند استفاده شده است. این ترکیب یک رابط کاربری قوی و کاربرپسند را فراهم می‌کند و از قابلیت‌های شی‌گرایی #C برای سازماندهی کد کارآمد استفاده می‌کند [۷و۸]. پلتفرم WPF، صفحه نمایش را با قابلیت‌های گرافیکی غنی بهبود می‌بخشد و امکان ارائه نرم‌افزار داده‌های رادار را فراهم می‌سازد. علاوه بر این، تطبیق‌پذیری #C ادغام با سایر سیستم‌ها را ساده می‌کند و آن را به گزینه‌ای ایده‌آل برای توسعه برنامه‌های رادار پیشرفته و بدون مرز PPI تبدیل می‌کند [۹]. هم‌افزایی بین شبیه‌سازی رادار سنتی و فناوری شبکه‌های عصبی نه تنها روش‌های آموزشی را پیشرفت می‌دهد، بلکه رویکرد آینده‌نگر به آموزش دفاع هوایی را نیز منعکس می‌کند [۱۰و۱۱]. این شبیه‌ساز که به طور خاص برای دانشجویان دانشگاه پدافند هوایی خاتم الانبیاء(ص) پیاده‌سازی گردیده است، به عنوان ابزاری ارزشمند برای تقویت مهارت‌های مورد نیاز برای بهره‌گیری مؤثر نمایشگرهای راداری PPI عمل می‌کند. با این حال، آنچه این پروژه را نسبت به سایر شبیه‌سازی‌های راداری متمایز می‌سازد،

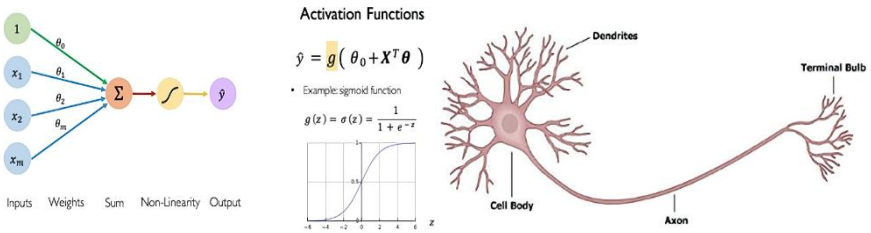
[1] Plan position indicator
[2] Windows Presentation Foundation

ترکیب نوآورانه آن از یک شبکه عصبی است که به اپراتورها امکان پیش‌بینی مکان‌های حمله را بر اساس سناریوهای شبیه‌سازی شده ارائه می‌دهد.
هسته اصلی این ویژگی نوآورانه یک شبکه عصبی چهار لایه است که به طور یکپارچه با شبیه‌ساز ادغام شده است. این شبکه عصبی شامل یک لایه ورودی، دو لایه پنهان و یک لایه خروجی است و هدف پیاده‌سازی آن پردازش اطلاعات حیاتی و ارائه دیدگاه وسیع‌تر به اپراتورها در مورد احتمال وقوع حمله در یک مکان مشخص طراحی شده است. لایه ورودی شبکه مذکور شامل سه ویژگی کلیدیست که عبارت‌اند از سرعت، جهت و فاصله، در حالی که لایه خروجی درصد احتمال از حمله قریب‌الوقوع را ارائه می‌دهد. انعطاف‌پذیری این پروژه به کاربر اجازه می‌دهد تا شبکه عصبی را با تعیین تعداد نورون‌ها در لایه‌های پنهان سفارشی‌سازی کند و تجربه یادگیری مناسبی را برای کاربر فراهم کند.
فرآیند آموزش از طریق اجرای کتابخانه Encog، یک مجموعه ابزار قدرتمند برای توسعه شبکه عصبی تسهیل می‌شود [۱۲].
یکی از ویژگی‌های برجسته نمایشگرهای راداری دیجیتال پیشرفته، ادغام داده‌های جغرافیایی دنیای واقعی برای افزایش صحت سناریوهای آموزشی است. در این شبیه‌ساز نیز اپراتورها می‌توانند مکان‌های خاص مورد علاقه را که بر اساس طول و عرض جغرافیایی مشخص شده‌اند در لیست مکان‌های حساس نرم‌افزار قرار داده و احتمال حمله به آن‌ها را توسط اهداف شبیه‌سازی شده و از طریق شبکه عصبی بصورت بصری مشاهده کنند. این رویکرد پویا برای ایجاد سناریو یک تجربه عملی را فراهم می‌کند، فرآیند یادگیری را غنی می‌سازد و دانشجویان حوزه پدافند را برای طیف گسترده‌ای از تهدیدات بالقوه آماده می‌کند.

۲-روش تحقیق

۲-۱-شبکه عصبی

شبکه عصبی یک مدل محاسباتی الهام گرفته شده از ساختار و عملکرد مغز انسان است که برای تشخیص الگوها، یادگیری از داده‌ها و پیش‌بینی یا تصمیم‌گیری طراحی شده است [۱۳]. شکل (۱) سمت چپ یک شبکه عصبی بیولوژیکی مغز و در سمت راست مدل مصنوعی منشأ گرفته از آن را نشان می‌دهد. این شبکه شامل گره‌های به هم پیوسته یا نورون‌های مصنوعی است که در لایه‌ها سازماندهی شده‌اند. سه نوع لایه اصلی عبارتند از: لایه‌ورودی، لایه‌های پنهان و لایه خروجی. از نظر فنی، یک لایه از مجموعه‌ای از نورون‌ها تشکیل شده است که یک بردار ورودی مشترک دارند اما به یکدیگر متصل نیستند. به طور معمول، نورون‌های موجود در یک لایه، همه از یک تابع فعال سازی استفاده می‌کنند.



شکل (۱): مدل‌سازی شبکه عصبی بیولوژیکی [۱۳]

۱. لایه ورودی: لایه اولیه داده‌های ورودی خام را دریافت می‌کند که نشان دهنده ویژگی‌های مسئله است. هر نورون در این لایه مربوط به یک ویژگی است و در مجموع بردار ورودی شبکه را تشکیل می‌دهند. در شکل (۱) $(x_1, .., x_d)$ بردار ویژگی‌های ورودی را نشان می‌دهند.

۲. لایه‌های پنهان: بین لایه‌های ورودی و خروجی، یک یا چند لایه پنهان می‌تواند وجود داشته باشد. هر نورون در یک لایه پنهان، ورودی لایه قبلی را از طریق اتصالات وزنی $(w_0, w_1, .., w_d)$ پردازش می‌کند و یک تابع فعال‌سازی را برای تولید خروجی اعمال می‌کند. وزن‌ها در طول آموزش برای بهینه‌سازی عملکرد شبکه تنظیم می‌شوند.

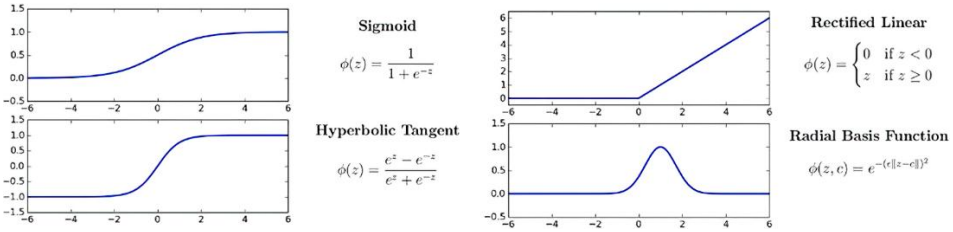
۳. لایه خروجی: لایه نهایی، خروجی شبکه را بر اساس الگوهای آموخته شده از داده‌های ورودی تولید می‌کند. تعداد نورون‌ها در این لایه به ماهیت کار بستگی دارد. به عنوان مثال، یک کار طبقه‌بندی باینری ممکن است یک نورون خروجی داشته باشد، در حالی که یک مسئله چند کلاسه ممکن است چندین نورون داشته باشد. رابطه (۱) خروجی شبکه عصبی شکل (۱) را پس از عبور از تابع فعال‌ساز g نشان می‌دهد [۱۴].

$$y(x) = g(w_0 + \sum_{i=1}^{d} w_i x_i) = g(\sum_{i=0}^{d} w_i x_i) = g(w^T x) \qquad (۱)$$

در (۱) W_0 بایاس[1] لایه ورودی است. بایاس در یک شبکه عصبی یک پارامتر اضافی است که به هر لایه اضافه می‌شود و به مدل اجازه می‌دهد تا آفست یا تغییر داده را بپذیرد. این به شبکه کمک می‌کند تا الگوهای پیچیده را بیاموزد و انعطاف پذیری را بهبود بخشد و از نمایش مؤثر و سازگاری با شرایط ورودی مختلف اطمینان حاصل کند.

۴. وزن‌ها و توابع فعال‌سازی: وزن‌ها نشان‌دهنده قدرت اتصالات بین نورون‌ها هستند و در طول فرآیند آموزش برای به حداقل رساندن تفاوت بین خروجی‌های پیش‌بینی شده و واقعی تنظیم می‌شوند. توابع فعال‌سازی غیرخطی بودن را معرفی می‌کنند و به شبکه اجازه می‌دهند تا روابط پیچیده در داده‌ها را بیاموزد. در شکل (۲) توابع فعال‌ساز رایج و پرکاربرد نشان داده شده‌اند.

[1] Bias



شکل (۲): توابع فعال رایج مورد استفاده در خروجی نورون‌های شبکه عصبی [۱۳]

۲-۱-۱- شبکه عصبی پرسپترون تک لایه

پرسپترون ساده‌ترین شکل یک شبکه عصبی و یک مفهوم اساسی در زمینه هوش مصنوعی و یادگیری ماشین است و به عنوان بلوک اصلی برای شبکه‌های عصبی مصنوعی پیچیده‌تر عمل می‌کند. پرسپترون توسط فرانک روزنبلات در سال ۱۹۵۷ توسعه یافت. اگرچه اصطلاح "تک لایه" یک لایه نورون را نشان می‌دهد، اما تعداد نورون‌ها در لایه خروجی را محدود نمی‌سازد. می‌توان چندین نورون در لایه خروجی وجود داشته باشد که هر کدام خروجی خود را بر اساس ویژگی‌های ورودی تولید می‌کنند. این به شبکه عصبی تک لایه اجازه می‌دهد تا طبقه‌بندی چند کلاسه را انجام دهد [۱۳].

بنابراین از طریق تعمیم شبکه عصبی تک خروجی رابطه (۱) می‌توان به شبکه عصبی تک لایه با چندین خروجی دست یافت. رابطه (۲) خروجی شبکه عصبی تک لایه با چندین خروجی را نشان می‌دهد [۱۳].

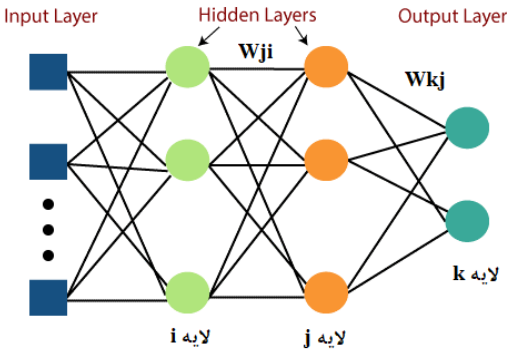
$$y_k(x) = g(\sum_{i=0}^{d} w_{ki} x_i) = g(w_k^T x) \qquad (۲)$$

۲-۱-۲- شبکه عصبی پرسپترون چند لایه

پرسپترون چندلایه، که اغلب به عنوان شبکه عصبی پیش‌خور نامیده می‌شود، کلاسی از شبکه‌های عصبی مصنوعی است که از لایه‌های متعددی از گره‌ها یا نورون‌های به هم پیوسته تشکیل شده است. برخلاف پرسپترون تک لایه، MLP [1] شامل یک لایه ورودی، یک یا چند لایه پنهان و یک لایه خروجی است [۱۵]. این معماری لایه‌ای شبکه را قادر می‌سازد تا روابط و الگوهای پیچیده در داده‌ها را بیاموزد. MLPها یک مورد خاص از شبکه‌های پیش‌خور هستند که در آن سه شرط اصلی

[1] Multilayer perceptron

برقرار است. این سه شرط عبارت‌اند از: نورون‌ها در لایه‌ها مرتب شده‌اند، حداقل یک لایه پنهان وجود دارد و هر لایه به طور کامل به لایه بعدی متصل است و هیچ اتصالی در لایه‌ها مجاز نیست.



شکل (۳): شبکه عصبی پرسپترون چند لایه [۱۵]

اگر روی یک نورون در یک لایه تمرکز کنیم، برای مثال نورون m را در لایه j ام فرض کنید، خروجی آن اکنون از طریق رابطه (۳) محاسبه می‌شود [۱۵]:

$$y_m(x) = g(\sum_{i=0}^{h} w_{mi}\varphi_i(x_i)) \qquad (۳)$$

در (۳) $\varphi_i(x)$ خروجی نورون i در لایه قبلی است. همچنین توجه داشته باشید که مجموع از $i = 0$ به $i = h$ می رسد و h اکنون تعداد نورون‌های لایه قبلی است.

تمام شبکه‌های عصبی را می‌توان به دو بخش تقسیم کرد: مرحله انتشار رو به جلو، جایی که اطلاعات برای محاسبه پیش‌بینی‌ها و خطا به جلو جریان می‌یابد. و مرحله انتشار به عقب، که در آن الگوریتم پس انتشار مشتقات خطا را محاسبه می‌کند و وزن‌های سیناپسی شبکه به روز رسانی می‌شوند.

- انتشار رو به جلو:

۱) مقادیر ورودی به نورون‌های لایه ورودی وارد می‌شود.

۲) هر ارتباط بین نورون‌ها با یک وزن مرتبط است. مقادیر ورودی در این وزن‌ها ضرب می‌شوند و از طریق یک تابع فعال سازی عبور می‌کنند.

۳) مجموع وزنی ورودی‌ها و بایاس‌ها، فعال شدن هر نورون در لایه‌های پنهان را تعیین می‌کند و در نهایت، لایه خروجی محاسبه می‌شود. این فرآیند لایه به لایه تکرار می‌شود تا خروجی نهایی از لایه خروجی به دست آید.

- انتشار به عقب (آموزش):

۱) در ابتدا یک تابع هزینه تعریف می‌شود که عبارت است از تفاضل خروجی پیش‌بینی شده (y_o) و مقدار واقعی خروجی (d)، برای مثال میانگین مربعات خطا برای رگرسیون یا آنتروپی متقابل با هدف طبقه‌بندی. براین اساس رابطه (۴) تابع هزینه را مشخص می‌کند.

$$L = \frac{1}{2}\sum_j^C (d_j - y_{oj})^2 = \frac{1}{2}\sum_j^C e_j^2 \qquad (۴)$$

در تابع هزینه C تعداد نورون در لایه خروجی و e_j خطای مرتبط با نورون در حال انجام محاسبات است.

۲) این خطا با استفاده از فرآیندی به نام پس انتشار از طریق شبکه به عقب منتشر می‌شود.

۳) وزن‌ها و بایاس‌ها بر اساس گرادیان خطا نسبت به هر وزن تنظیم می‌شوند و شبکه را برای عملکرد بهتر، بهینه می‌کنند. به عبارتی دیگر از تابع هزینه نسبت به وزن‌ها مشتق گرفته شده و در جهت شیب منفی خطا حرکت می‌شود تا به مقدار حداقل رسیده شود.

$$w_{ji}^{new} = w_{ji}^{old} - \eta \nabla L \qquad (۵)$$

در (۵) W_{ji} وزن‌های سیناپسی بین لایه jام و iام هستند، η نرخ یادگیری یا به عبارتی دیگر اندازه گام در فرآیند به روزرسانی وزن‌ها در گرادیان نزولی است. ∇L گرادیان تابع هزینه نسبت به وزن‌ها است.

مطابق با رابطه (۶) ثابت می‌شود که گرادیان نزولی خطا نسبت به وزن‌ها برابر است با:

$$-\eta \nabla L = \eta . \delta_j . y_i \qquad (۶)$$

در (۶) y_i خروجی نورون لایه قبلی یا همان لایه iام است و به عبارتی دیگر ورودی نورون لایه فعلی یا همان لایه jام است.

بنابراین با ترکیب روابط (۵) و (۶)، رابطه (۷) که برای به روزرسانی وزن‌های سیناپسی استفاده می‌شود، مستخرج می‌شود.

$$w_{ji}^{new} = w_{ji}^{old} + \eta . \delta_j . y_i \qquad (۷)$$

لایه خروجی δ از طریق رابطه (۸) محاسبه می‌شود. در صورتی که شبکه عصبی دارای یک خروجی باشد، δ_k یک ماتریس یک در یک خواهد بود.

$$\delta_k = e_k . g'(\sum_{j=0}^{h} w_{kj} y_j) \qquad (۸)$$

اما بدلیل عدم امکان محاسبه خطا در لایه پنهان، برای محاسبه δ در لایه‌های پنهان از رابطه (۹) استفاده می‌شود. بنابراین در لایه jام به ازای هر نورون باید یک δ محاسبه شود.

$$\delta_j = g'(\sum_{i=0}^{h} w_{ji} y_i).\sum \delta_k w_{kj} \qquad (۹)$$

۴) فرآیند انتشار به جلو و عقب به طور مکرر در طول آموزش تکرار می‌شود تا زمانی که مدل به حالتی همگرا شود که خطا به حداقل برسد.

۳-۱-۲- پیاده‌سازی شبکه عصبی در سی‌شارپ

به منظور پیاده‌سازی شبکه عصبی در این تحقیق از کتابخانه Encog[1] استفاده شده است. Encog یک چارچوب یادگیری ماشین منبع باز برای جاوا و دات‌نت است که قابلیت‌های پیشرفته‌ای را برای شبکه‌های عصبی و همچنین پشتیبانی از سایر الگوریتم‌های یادگیری ماشین را فراهم می‌کند. در حالی که پیاده‌سازی اولیه در جاوا انجام می‌شود، یک نسخه دات‌نت از Encog نیز وجود دارد که به توسعه دهندگان اجازه می‌دهد تا از قابلیت‌های آن در C# استفاده کنند.

۱) **معماری شبکه عصبی Encog**: این کتابخانه از شبکه‌های عصبی پیش‌خور[2] پشتیبانی می‌کند که رایج‌ترین نوع شبکه‌های عصبی مصنوعی هستند. این شبکه‌ها از لایه‌هایی از نورون‌های به هم پیوسته تشکیل شده‌اند و در آن اطلاعات در یک جهت از لایه ورودی به لایه خروجی جریان دارد. در این تحقیق نیز از کتابخانه Encog تحت عنوان شبکه عصبی پرسپترون چند لایه که یک شبکه عصبی پیش‌خور محسوب می‌شود، استفاده شده است. این کتابخانه همچنین از شبکه‌های عصبی مکرر (RNN)[3] پشتیبانی می‌کند، که به اطلاعات اجازه می‌دهد تا مدتی باقی بمانند. این روش برای کارهایی که شامل داده‌های متوالی هستند مفید است [۱۶].

۲) **توابع فعال‌سازی:** Encog از توابع فعال‌سازی مختلفی پشتیبانی می‌کند که غیرخطی بودن را به شبکه عصبی وارد می‌کند و آن را قادر می‌سازد تا روابط پیچیده در داده‌ها را فرابگیرد. توابع فعال‌سازی متداول شامل سیگموئید، مماس هذلولی (tanh) و واحد خطی اصلاح شده (ReLU) است [۱۶].

۳) **الگوریتم‌های آموزشی:** این کتابخانه الگوریتم‌های آموزشی متنوعی را برای شبکه‌های عصبی ارائه می‌کند که به کاربران این امکان را می‌دهد متناسب با مشکل خاص خود

[1] Enhanced Cognition Framework
[2] Feedforward Networks
[3] Recurrent Networks

الگوریتم آموزشی را انتخاب کنند. الگوریتم‌های آموزشی متداول شامل پس انتشار[1]، انتشار ارتجاعی (Rprop)[2]، انتشار سریع و الگوریتم‌های تکاملی مختلف مانند الگوریتم‌های ژنتیک شبیه‌سازی شده است [۱۷].

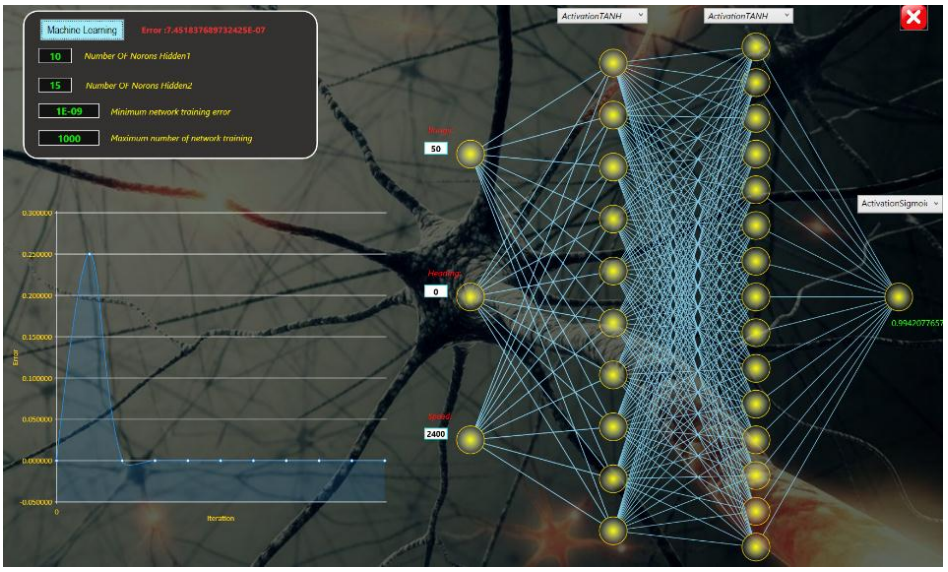
۴) **پیش پردازش داده‌ها:** این کتابخانه ابزارهایی را برای پیش پردازش و نرمال‌سازی داده‌ها قبل از آموزش شبکه عصبی فراهم می‌کند. این امر برای دستیابی به عملکرد بهینه و همگرایی در طول فرآیند آموزش بسیار مهم است [۱۷].

۴-۱-۲-ساختار شبکه عصبی نمایشگر PPI

در این تحقیق یک شبکه عصبی پرسپترون با دو لایه پنهان، یک لایه ورودی و یک لایه خروجی با استفاده از کتابخانه Encog پیاده‌سازی شده است. با استفاده از این شبکه عصبی، مکان مورد حمله اهداف به همراه درصد آن پیش‌بینی می‌شود. ورودی‌های شبکه عبارت‌اند از، سه پارامتر فاصله، جهت و سرعت. خروجی شبکه نیز درصد احتمال حمله به تمامی مکان‌های حساس مورد نظر در فایل اکسل می‌باشد. ویژگی‌ها و قابلیت‌های شبکه عصبی پیاده‌سازی شده در شبیه‌ساز نمایشگر رادار عبارتند از:

۱) شبکه عصبی ۴ لایه (دولایه پنهان)؛
۲) قابلیت تعیین تعداد نورون‌های لایه‌های پنهان؛
۳) قابلیت تعیین تعداد بار آموزش؛
۴) امکان تعیین حداقل خطا در آموزش؛
۵) نمایش ساختار گرافیکی شبکه عصبی به منظور فهم بیشتر اپراتور؛
۶) نمایش نمودار محاسبه خطا پس از هر بار آموزش؛
۷) امکان تغییر داده‌های آموزش از طریق فایل اکسل در مسیر پروژه؛
۸) امکان تعیین نوع تابع فعال در هر لایه؛
۹) امکان تغییر (حذف، اضافه) مکان‌های حساس کشور از طریق فایل اکسل در مسیر پروژه؛
۱۰) آموزش سریع و بهینه؛ و
۱۱) اجرای شبکه عصبی در رشته[3] جهت افزایش سرعت محاسبات و جلوگیری از کند شدن عملکرد نمایشگر.

[1] Backpropagation
[2] Resilient propagation
[3] Thread



شکل (۴): رابط کاربری شبکه عصبی پیاده‌سازی شده در شبیه‌ساز نمایشگر PPI

مطابق شکل (۴) اپراتور می‌تواند تعداد نورون‌های لایه پنهان اول و دوم را مطابق نیاز تغییر دهد. همچنین می‌تواند حداقل خطای شبکه را تنظیم کند و در هر مرحله که خطای آموزش شبکه به این مقدار برسد آموزش متوقف می‌شود. در صورتی که خطای آموزش هیچگاه به این مقدار نرسد پس از رسیدن به بیشینه تعداد تکرار آموزش که آن نیز قابل تنظیم است متوقف می‌شود.

آموزش شبکه عصبی پیاده‌سازی شده از طریق فایل اکسل انجام می‌شود. در این فایل داده‌های آموزش سه پارامتر فاصله، جهت و سرعت که در واقع سه ویژگی شبکه مورد نظر هست وجود دارد که توسط کاربر قابل تغییر است.

خروجی شبکه عصبی (درصد احتمال حمله) به ازای هر داده آموزش در فایل اکسل مطابق جدول (۱) بایستی تعیین گردد. سه ویژگی ورودی این شبکه عصبی بدلیل پراکندگی بسیار، قبل از ورود به مرحله آموزش، نرمالایز[1] می‌شوند. از انتخاب بیش از حد داده در آموزش نیز خودداری شده تا از بوجود آمدن مشکل برازش بیش از حد[2] در شبکه جلوگیری شود.

جدول (۱): بخشی از داده‌های آموزش شبکه عصبی در فایل اکسل نرم‌افزار

احتمال حمله (۱-۰)	سرعت هدف برحسب کیلومتر بر ساعت(speed)	جهت حرکت هدف نسب به مکان برحسب درجه(Heading) (۱۸۰-۰)	فاصله هدف از مکان برحسب کیلومتر(Range)
۱	۲۴۰۰	۰	۵۰

[1] Normalize
[2] Overfitting

۰.۹۸	۲۰۰۰	۰	۵۰
۰.۹۶	۱۶۰۰	۰	۵۰
۰.۹	۴۰۰	۰	۵۰
۰.۸۸۲۵	۵۰	۰	۵۰
۰.۸۶	۲۴۰۰	۱۰	۵۰
۰.۳	۲۴۰۰	۱۰	۵۰۰
۰.۷۲	۲۴۰۰	۲۰	۵۰
۰.۱۵	۲۴۰۰	۱۸۰	۵۰
۰.۸	۲۰۰۰	۱۰	۵۰
۰.۶۶	۱۶۰۰	۲۰	۵۰
۰.۹۲	۲۴۰۰	۰	۷۰
۰.۵	۲۴۰۰	۰	۵۰۰
۰.۷۲	۲۰۰۰	۱۰	۶۰

تعیین موقعیت‌های حساس کشور از طریق فایل اکسل تحت عنوان ImportantPlace انجام می‌شود که در آن برخی از مکان‌های حساس و مهم کشور به همراه مختصات جغرافیایی آن‌ها ثبت شده است. این مکان‌ها شامل نیروگاه‌ها، سدها، فرودگاه‌ها و مراکز اتمی کشور است. این لیست قابل تنظیم و تغییر است و اپراتور می‌تواند مکان‌های حساس مورد نظر خود را در این فایل اکسل ثبت کند.

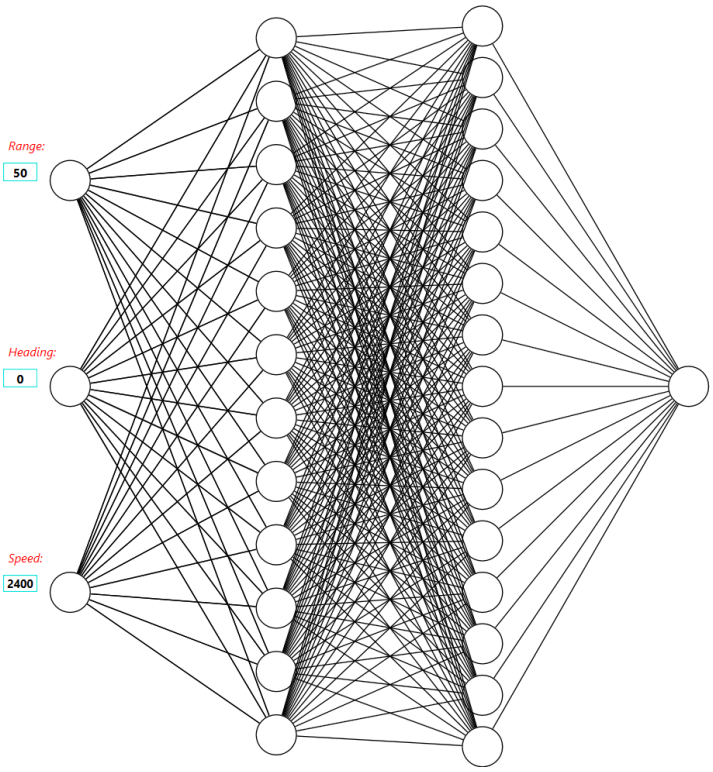
شبکه عصبی احتمال حمله به این لیست را محاسبه می‌کند و باتوجه به سه ویژگی سرعت، فاصله و جهت هدف، ۵ مکان از محتمل‌ترینِ این مکان‌ها را به همراه درصد احتمال حمله به آن‌ها را به کاربر نمایش می‌دهد.

جدول (۲): بخشی از مختصات جغرافیایی مکان‌های مهم و حساس کشور در فایل اکسل ImportantPlace

نام مکان مهم	طول جغرافیایی	عرض جغرافیایی
نیروگاه حرارتی توس	۵۹.۴۱۱۶	۳۶.۴۵۲۹
نیروگاه شهید منتظری	۵۱.۴۹۶۶	۳۲.۷۹۷۵
نیروگاه شهید رجایی	۵۰.۳۰۴۳	۳۶.۱۵۹۸
نیروگاه سیکل ترکیبی کرمان	۵۶.۷۹	۳۰.۲۰۹۴
پالایشگاه آبادان	۴۸.۲۷۱۲۴۳۶۳	۳۰.۳۶۴۸۸۶۸۵
پالایشگاه شیراز	۵۲.۶۵۹۴۷۲۴۳	۲۹.۷۵۹۷۹۷۸۴
تأسیسات راکتور آب سنگین اراک	۴۹.۲۴۱	۳۴.۳۷۱۲
کارخانه غنی سازی سوخت فردو	۵۰.۹۹۶	۳۴.۸۸۴۷
انرژی اتمی اصفهان	۵۱.۷۸۸۸	۳۲.۵۸۰۷

۳-نتایج و بحث

شبکه عصبی تنظیم شده جهت بررسی نتایج مدل پیش‌بین مطابق شکل (۵) دارای ۱۰ نورون در لایه پنهان اول و ۱۵ نورون در لایه پنهان دوم است.

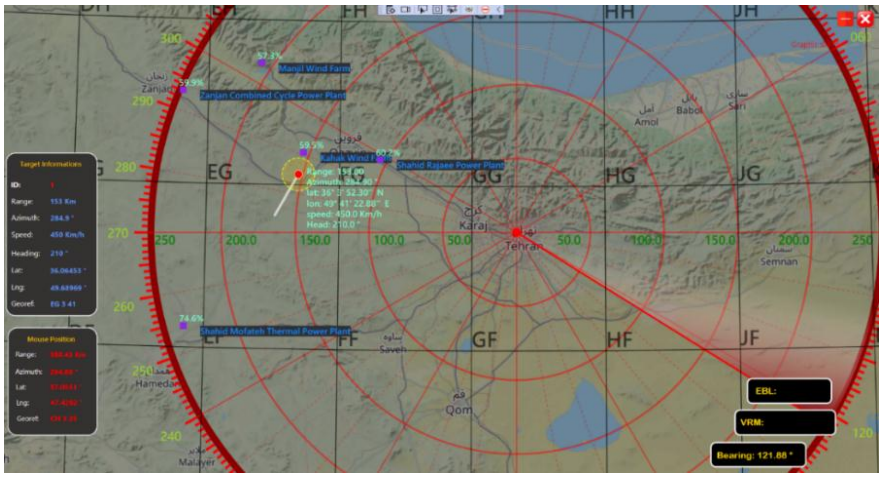


شکل (۵): گراف شبکه عصبی مورد آزمایش

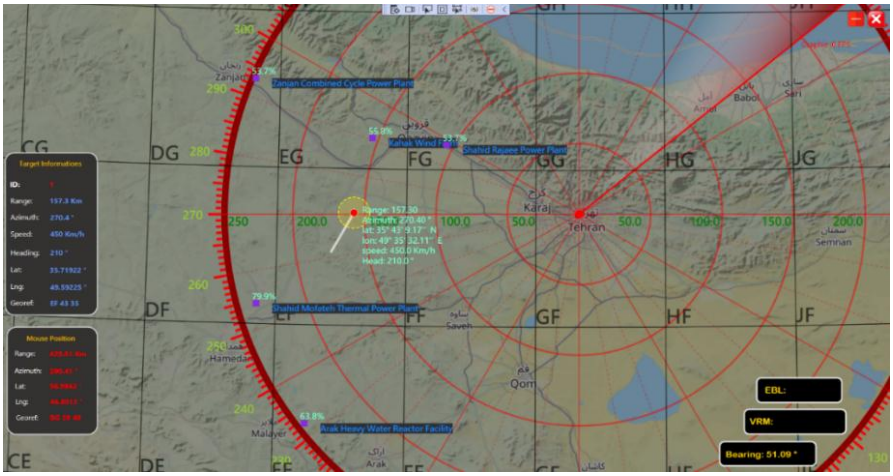
در شبکه فوق تعداد بار آموزش ۵۰۰۰ و حداقل خطا 1E-9 تنظیم شده‌اند. حال برای یک هدف در دو موقعیت جغرافیایی، نتایج مدل پیش‌بینی طبق جدول (۳) مشخص گردیده است.

جدول (۳): مقایسه داده‌های خروجی شبکه عصبی به ازای دو موقعیت جغرافیایی یک هدف

موقعیت جغرافیایی اول (36° 3' 52.3" N , 49° 41 22.88" E)		موقعیت جغرافیایی دوم (35° 43' 9.17" N , 49° 35 32.11" E)	
مکان مورد تهاجم	احتمال حمله	مکان مورد تهاجم	احتمال حمله
نیروگاه بادی منجیل	۵۷.۳	نیروگاه سیکل ترکیبی زنجان	۵۳.۷
نیروگاه بادی خدک	۵۹.۵	نیروگاه شهید رجایی	۵۳.۷
نیروگاه سیکل ترکیبی زنجان	۵۹.۹	نیروگاه بادی خدک	۵۵.۸
نیروگاه شهید رجایی	۶۰.۲	تأسیسات راکتور آب سنگین اراک	۶۳.۸
نیروگاه حرارتی شهید مفتح	۷۴.۶	نیروگاه حرارتی شهید مفتح	۷۹.۹



شکل (۶): نمایش ۵ مکان با بیشترین درصد احتمال حمله در موقعیت جغرافیایی اول



شکل (۷): نمایش ۵ مکان با بیشترین درصد احتمال حمله در موقعیت جغرافیایی دوم

آنچه که مشخص است با تغییر موقعیت و مختصات جغرافیایی هدف، علاوه بر تغییر درصد احتمال هر مکان مورد تهدید، اولویت مکان‌ها نیز تغییر کرده است. در مختصات دوم نیروگاه بادی منجیل حذف و تأسیسات راکتور آب سنگین اراک اضافه می‌گردد.

البته این دو نتیجه با گذشت زمان و تغییر جابه‌جایی قابل ملاحظه‌ای ثبت گردیده و لازم به ذکر است خروجی شبکه عصبی با هر تغییر جزیی هدف به روزرسانی شده و درصد تغییرات به طور پیوسته محاسبه می‌گردد.

٤-تشکر و قدردانی

از دفتر پژوهشکده دانشگاه پدافند هوایی خاتم الانبیاء(ص) بابت پیگیری‌ها و هماهنگی‌های لازم جهت برگزاری جلسات خبرگی پژوهش حاضر و نیز کلیه اندیشمندان و پژوهشگرانی

که در خلال تحقیق خالصانه دیدگاه‌ها و نقطه نظرات علمی و کارشناسی خود را ارائه نمودند، تشکر و قدردانی می‌گردد.

۵-تعارض منافع

نویسنده(گان) اعلام می‌دارند که در مورد انتشار این مقاله تضاد منافع وجود ندارد. علاوه بر این، موضوعات اخلاقی شامل سرقت ادبی، رضایت آگاهانه، سوء رفتار، جعل داده‌ها، انتشار و ارسال مجدد و مکرر توسط نویسندگان رعایت شده است.

٦-دسترسی آزاد

این نشریه دارای دسترسی باز است و اجازه اشتراک (تکثیر و بازآرایی محتوا به هر شکل) و انطباق (بازترکیب، تغییر شکل و بازسازی بر اساس محتوا) را می‌دهد.

۷-نتیجه‌گیری

در این مقاله پیاده‌سازی شبکه عصبی پرسپترون چهار لایه جهت تخمین مسیر آتی اهداف و ترکیب آن با شبیه‌ساز نمایشگر PPI با موفقیت انجام شد. همانطور که اشاره شد جهت پیاده‌سازی و برنامه‌نویسی این شبیه‌ساز از زبان C# و پلتفرم WPF بهره گرفته شده است و هدف اصلی، آموزش کاربر رادار حوزه پدافند با فناوری‌های نوین در راستای تخمین و شناسایی تهدیدات بالقوه است. به طبع توسعه و طراحی این نرم‌افزار کمک شایانی در آموزش و افزایش بهره‌وری اپراتور رادار و تسلط به شناسایی حملات احتمالی به مکان‌های حساس کشور ایران را به همراه دارد. شبکه عصبی توصیفی در این مقاله از کتابخانه Encog بهره گرفته است و با ادغام این فناوری با یک نمایشگر PPI دیجیتال مورد ارزیابی و آزمون قرار گرفته که همانطور که در مقاله اشاره شد با موفقیت تمامی آزمون‌های خود را پشت سر گذاشت. لذا شایان ذکر است مهمترین ویژگی این شبیه‌ساز، قابلیت آموزش شبکه عصبی و تعریف سناریوهای متفاوت توسط دانشجویان متخصص در این حوزه است. بنابراین امکان تعیین سناریوهای متفاوت به منظور تخمین مسیرآتی اهداف فراهم گردیده است. همچنین امکان تعریف مکان‌های حساس و دسته‌بندی آنها نیز توسط کاربر فراهم شد. شبیه‌ساز پیاده‌سازی شده با تلفیق داده‌های جغرافیایی دنیای واقعی، خروجی شبکه عصبی را بصورت بصری بر روی صفحه نمایش رادار نشان داده که سبب درک بهتر اپراتور شده و صحت سناریوهای طراحی شده را به تصویر میکشد. برخی از نتایج این شبیه‌ساز در بخش‌های مختلف این مقاله نشان داده شده است. مهم‌ترین ویژگی‌های این نرم‌افزار شبیه‌ساز نمایشگر PPI که در این مقاله به اثبات رسید و آزمون‌های خود را با موفقیت سپری نموده عبارتنداز: پیاده‌سازی و تلفیق شبکه عصبی چهار لایه منبع باز با قابلیت‌های فراوان(تنظیم تعداد نورون‌ها، تعیین نوع آموزش، تعیین نوع توابع فعال و.....)، محاسبه و نمایش ۵ مکان از محتمل‌ترین مکان‌های حساس مورد تهاجم، امکان تعیین و

دسته‌بندی مکان‌های حساس توسط اپراتور و تلفیق نمایش خروجی شبکه عصبی با داده‌های جغرافیایی دنیای واقعی.

منابع

[1] H, Mohseni.; M, Najafzadeh.; M, zarie.; A, Saravania.; S, Zare. "Calculation of the Combined Threat of Air Targets Using Neuro-Fuzzy Systems," 2022, 69-78.

[2] Long, T.; Liang, Z.; Liu, Q. "Advanced technology of high-resolution radar: Target detection, tracking, imaging, and recognition," Sci. China Inf. Sci. 2019, 62, 1–26.

[3] Wang, Yinhan, et al. "Quick intention identification of an enemy aerial target through information classification processing," *Aerospace Science and Technology* 132, 2023, 108005.

[4] Cai, Xiuzhang, Michael Giallorenzo, and Kamal Sarabandi. "Machine learning-based target classification for MMW radar in autonomous driving," IEEE Transactions on Intelligent Vehicles 6.4: 2021, 678-689.

[5] Xu, Baoxiong, et al. "High-accuracy target tracking for multistatic passive radar based on a deep feedforward neural network," Frontiers of Information Technology & Electronic Engineering 24.8, 2023, 1214-1230.

[6] Goodall, Finn, and Bashar I. Ahmad. "Adaptation of multi-target tracker using neural networks in drone surveillance radar," IEEE Radar Conference (RadarConf23). IEEE, 2023.

[7] Adam, N. "Windows Presentation Foundation Unleashed," Sams Publishing. 2007.

[8] Grifoni, D.; MacDonald, B. "Pro WPF in C# 2010: Windows Presentation Foundation in .NET," 2010.

[9] Noyes, R.; "Windows Presentation Foundation 4.5 Cookbook," Packt Publishing. 2020.

[10] Q, Jin.; H, Wang.; K, Yang. "Radar emitter identification based on EPSD-DFN," IEEE 3rd Adv. Inf. Tech. Electronic and Automation Control Conference (IAEAC), Chongqing, 2018, pp. 360-363.

[11] J, Matuszewski.; K. Sikorska-Łukasiewicz, "Neural network application for emitter identification," 18th Int. Radar Symposium (IRS), Prague, 2017, pp. 1-8.

[12] J, Heaton. "Programming Neural Networks with Encog3 in Java," Missouri; Heaton Research, 2011.

[13] V, Alessandro.; et al., "eds. Artificial Neural Networks and Machine Learning-ICANN 2012," 22nd Int. Conference on Artificial Neural Networks, Lausanne, Proceedings, Part II, Switzerland, 2012, Vol. 7553. Springer.

[14] J, Singh.; R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," 2019 3rd Int. Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 35-40.

[15] A, Luis., "Multilayer perceptrons," Handbook of Neural Computation. CRC Press, 2020. C1-2.

[16] J, Heaton,. "Programming neural networks with Encog 2 in Java," Heaton Research, Inc., 2010.

[17] L, Igor,. "Artificial Neural Networks with Java: Tools for Building Neural Network Applications," Apress, 2019.